

S.E.V. Solar Extended Vehicle



EEL 4915

Senior Design II

Group #4

Hamed Alostath

Daniel Grainger

Frank Niles

Sergio Roig

**University of Central
Florida**

Dr. S. Richie

Fall 2011

Table of Contents

S.E.V.

1. Executive Summary	1
2. Initial Technical Content	2
2.1 Motivation and Goals.....	2
2.2 Project Requirements and Specifications.....	3
2.3 Market Research.....	3
3. Airframe Research	4
3.1 Introduction.....	4
3.2 Specifications.....	6
3.3 Research.....	6
3.3.1 Apprentice 15e.....	6
3.3.2 Parkzone Radian.....	7
3.3.3 The ProJet RQ-11.....	8
3.4 Functionality.....	9
3.5 Motors.....	11
3.5.1 Motor Research.....	11
3.6 Servos.....	12
3.6.1 Servo Research.....	12
3.7 Electronic Speed Controllers.....	13
3.7.1 ESC Research.....	13
3.8 Integration	14
4. Printed Circuit Boards	15
4.1 Circuit Board Requirements.....	15
4.2 PCB Vendors.....	16
4.3 PCB Specifications.....	18
4.4 Board Layout.....	19
4.5 PCB Testing.....	21
4.6 PCB Installation.....	21
5. Microcontroller	23
5.1 Microcontroller Requirements.....	23
5.2 Comparison and Selection.....	24
5.2.1 ATmega1280.....	24
5.2.2 ATmega328.....	25
5.2.3 ATtiny45.....	25
5.3 Microcontroller Integration.....	27
6. Sensors: Roll, Pitch, Yaw	30
6.1 Sensors.....	30
6.2 Gyro.....	30
6.2.1 Basic Requirements.....	31

6.2.2	Integration.....	32
6.3	Accelerometer.....	33
6.3.1	Basic Requirements.....	34
6.3.2	Integration.....	35
6.4	Barometric Pressure Sensor.....	36
6.4.1	Basic Requirement.....	36
6.4.2	Integration.....	37
7.	Communication.....	38
7.1	Machine to Machine Communications.....	38
7.1.1	Data Collection.....	38
7.1.2	Data Transmission.....	39
7.1.3	Revision.....	39
7.1.4	Response.....	39
7.2	Global Positioning System (GPS)	39
7.2.1	GPS Requirements/Features.....	40
7.2.2	GPS Comparison and Selection.....	40
7.2.3	Interface Design.....	41
7.3	ZigBee.....	42
7.3.1	Basic Requirements.....	43
8.	Power Systems.....	44
8.1	General Overview.....	44
8.2	Solar Cells.....	45
8.2.1	Basic Requirements.....	45
8.2.2	Selection.....	46
8.2.3	Interface Design.....	48
8.3	Battery.....	49
8.3.1	Basic Requirements.....	49
8.3.2	Selection.....	49
8.3.2.1	LiPo Battery Packs.....	50
8.3.2.2	LiPo Hybrid Battery Packs.....	50
8.3.3	Interface Design.....	51
8.4	Maximum Power Point Tracker (MPPT)	52
8.4.1	Charging Circuit.....	52
8.4.2	Basic Requirements.....	53
8.4.3	Design.....	54
8.4.4	Design Integration.....	58
9.	Software Design.....	58
9.1	Introduction.....	58
9.2	General Breakdown of ArduPilot Code.....	59
9.3	Altitude Heading Reference System.....	64
9.4	PC Software.....	65
9.4.1	ArduPilot Mission Planner.....	65
9.4.2	Ground Control Station.....	67

9.4.3 Simulator.....	68
10. Hardware Design.....	68
11. Design Summary of Hardware and Software.....	86
11.1 Block Diagram.....	99
12. Prototyping.....	100
12.1 Construction.....	100
13. Project Prototype Testing.....	106
13.1 Introduction.....	106
13.2 Testing.....	106
13.3 Rules & Regulations.....	108
13.4 Test Facilities.....	108
13.5 Flight Path.....	109
14. Administrative Content.....	109
14.1 Budget and Financing.....	109
14.2 Milestones.....	110
14.3 Milestone Chart.....	111
15. Senior Design II Modifications.....	114
15.1 Motor Mount.....	114
15.2 Motor.....	114
15.3 Electronic Speed Controller.....	116
15.4 Circuit Boards.....	116
15.4.1 Autopilot Circuit Board.....	117
15.4.2 Charging-Circuit Board.....	117
15.5 Sensors.....	118
15.6 Software.....	120
16. User Manual.....	121
Appendices.....	i
A. Bibliography.....	ii
B. Abbreviations.....	iii
C. Image Use Permissions.....	iv
D. Table of Figures.....	iv
E. Table of Tables.....	xii

1.0 Executive Summary

The unmanned aerial vehicle (UAV) has many real world applications but most of all it has significant military applications. UAV's are remotely piloted or self-piloted aircraft that can carry cameras, sensors, communications equipment, or other payloads. The military interests lie in the fact that they need vehicles for reconnaissance and surveillance missions. With such capabilities with unmanned aerial flight, it would in turn reduce the military's risk of human loss. Therefore, it would be able to take on more dangerous mission over hostile areas. UAVs can also be used in civil applications, such as firefighting, surveillance of pipelines and rescue type of operations over deserts and forests. Many civilian contractors are also looking at UAV's for "dull, dirty, or dangerous" type of jobs. With the ever-expanding role of the United States military in conflicts around the world, one thing that has become apparent is the importance of technological solutions that provide real-time, aerial reconnaissance to protect the life of the war fighter on the ground. The ability of these solutions to be mobile, rapidly deployable from anywhere and provide extended periods of surveillance is of the utmost importance. At this time, the most utilized solutions have been small, hand-launched UAVs that can be carried by ground support personnel. This project will focus on such a retrofit to the AeroVironment (AV) RQ-11 Raven B, seen in Figure 1. This aircraft is quickly becoming the standard for small UAVs, being deployed around the world by both the United States Marine Corps and Army for reconnaissance missions. For the purposes of this project, we are going to use a commercial grade airplane commonly known as "PROJET RQ-11 Raven" for our senior design project.

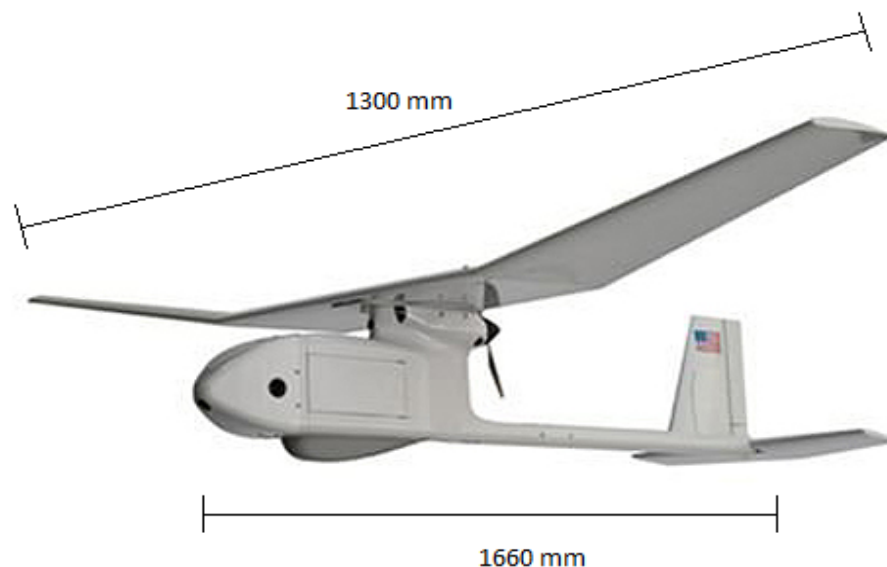


Figure 1 – RQ-11 Raven B

Reprinted with permission of AeroVironment Inc - pending

In addition to the military purposes, our Solar Extended Vehicle design is an attempt to harness solar power technology to extend the battery-life for UAVs. This idea of using solar power for UAVs is a relatively new idea. The main market of UAVs is for military purposes and there are over 50 vendors that produce these types of aircraft. While some attempts have been made, currently there are no solar-powered UAVs that are mass-produced. By creating, an efficient solar powered system at a relatively low cost, our Solar Extended Vehicle design may help pave the way for revolutionizing the solar powered aircraft.

Chapter 2: Initial Technical Content

2.1 Motivation and Goals

The group motivation for this project is to experience solar power systems, linear controls, become more efficient at programming. We, the members of group # 4, are to design a solar cell glider that is able to fly to a specific maximum elevation depending on the solar energy and the batteries that are used as power sources. Once it reaches that maximum elevation, which is programmed on the GPS, the solar cell starts charging batteries while the solar cell aircraft glides its way around until it reaches a minimum elevation. While it is gliding, the glider is designed to be able to survey the area by taking multiple pictures accompanied by the time and the position coordinates.

In words, it seems simple; however, here are some issues we might face along the way. The number one concern is the power to weight ratio; it has to be light and tough at the same time. We are talking about solar cells, batteries, electric components such as microprocessor, GPS, transmitter, receiver, servomotors, and the list goes on and on; all of that should be protected by a tough airframe to be able to fly and land safely without causing any damages. Solar cell sailplanes are proven to have the advantage for its flight period over military airplanes and commercial airplanes since it does not need to land for fueling unlike the military fighters and the commercial airplanes that have to land to refuel. On the other hand, its disadvantages fall in two categories speed and weight. The solar cell airplanes typically are not fast as the military fighters and cannot carry a lot of weight as the commercial airplanes; thus, the solar cell aircrafts are mostly unmanned systems. Described below is a list of generalized goals that our group devised during the initial planning phase of our project. These are the main goals that helped us narrow down the features of our project and also provided inspiration for some of the more advanced functions.

1. Autonomous
 - a. Follows a pre-programmed flight path via a GPS input
 - b. Automatically adjust for altitude and heading
 - c. Microprocessor that controls the rudder and elevator
 - d. Return to starting after battery reaches limit
2. Power Consumption

- a. Maintain power consumption as low as possible by integrating a power saving electric dc motor and using ultra low voltage microprocessor.
 - b. Integration of flexible thin cell solar cells
 - c. Solar powered on board battery charging system
3. Easy to use if manual control is needed

2.2 Project Requirements and Specifications

Once we had discussed the goals for the project, we developed a list of objectives that provided even more specific milestones for our design. The list of objectives allowed us to research more in-depth aspects of the project. Our objectives were laid out to serve as a broad array of subjects that we could implement on our project.

- Flight time on solar extended of ~1hr
- Wingspan between 4 ~7 ft
- Wing area of at least 400 sq in
- Plane weight of ~ 1000g
- Fully powered electric brushless motor 1020 RPM/Volt
- Self-Contained Lithium-ion polymer 1300mAh 3s ~ 2200mAh 3s battery pack
- 60-Amp ESC

2.3 Market Research

The capabilities of unmanned aerial vehicles continue to expand with the advent of new technologies. Recently, UAVs proved their combat worthiness in operations over Iraq and Afghanistan. These successes only increase the demand for the limited UAV resources. The unmanned aerial vehicle sector has been the most dynamic growth sector in aerospace industry. According Market research firm Forecast International claims that the total UAV market including air vehicles, ground control equipment and payloads are expected to be worth \$13.6 billion through 2014 in the United States alone. The U.S. military UAV market is projected to grow at a CAGR of 10% between 2010 and 2015, says, "U.S. Military Unmanned Aerial Vehicles (UAV) Market Forecast 2010-2015," a new research report from Market Research Media said. The report finds that the U.S. military UAV market will generate \$62 billion revenues over the period 2010-2015.

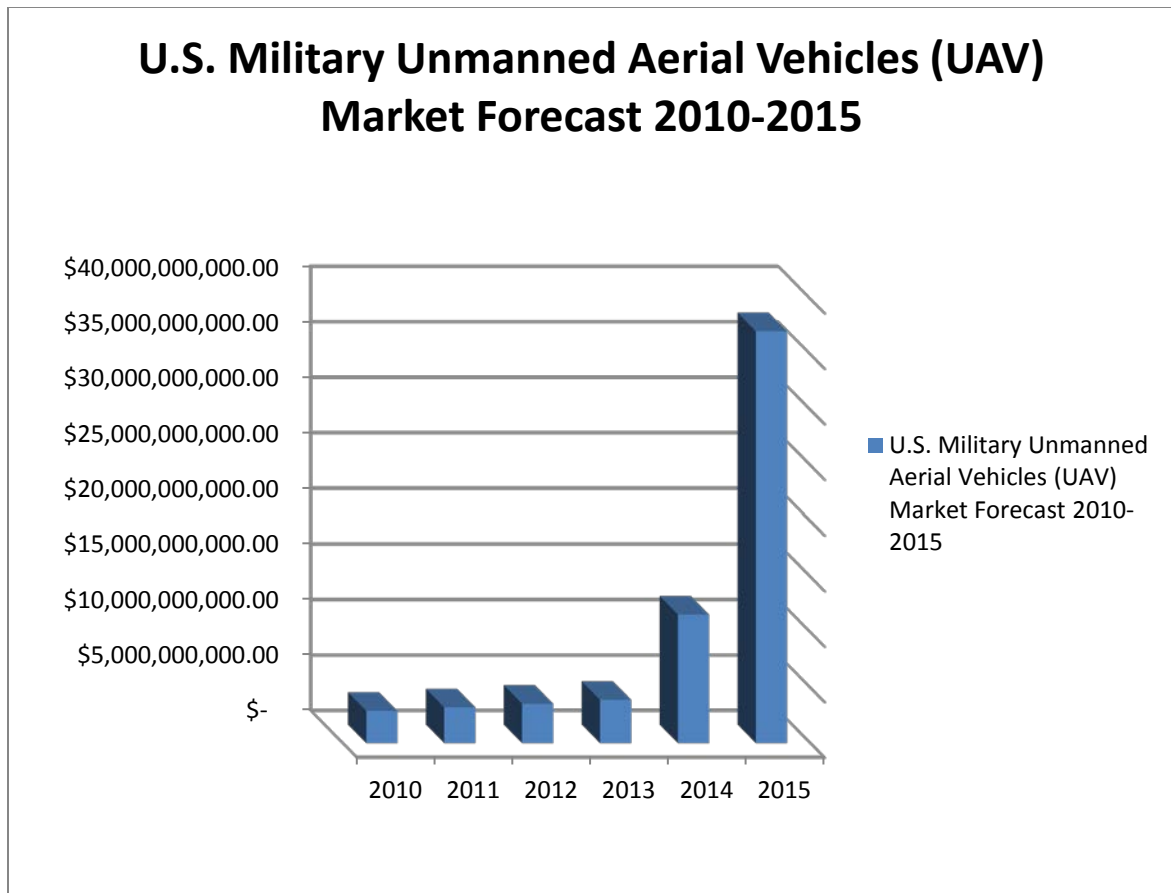


Figure 2.3.1 – Market Forecast 2010-2015

Based on our initial research of similar projects we found that several other students from various universities in the United States have attempted to create an unmanned vehicle. For instance, a student senior design group in Purdue University attempted a project named “Continuous Area Coverage via Fixed-Wing Unmanned Aerial Systems.” The project consisted of a UAV that housed a payload of a video camera, a thermal imaging camera, and a chemical detector. The goals of this project was to have the UAV fly in a five-mile radius and use the thermal imaging camera to detect specific heat signatures.

Chapter 3: Airframe Research

3.1 Introduction

The airframe research section relates to the initial research steps that were taken shortly after our project was chosen. This research began by selecting a plane to use as the frame of our S.E.V. design. We started by looking at RC planes and then later decided to choose an airframe that more closely resembled an UAV. Once we had chosen a hollow frame, we needed to research motors that would be applicable to our project. The motor would be one of the most significant components of the design since it would generate the

power for the entire system. Next, we needed to find servos that could be used for the minor mechanical functions. The last component is the electronic speed controller. The ESC would enable the motor and servos to be controlled from a single source. The ESC would serve as the link between the remote control and the plane. All of these components operate as the core system of the plane.

Our design will be a plane that will use a primary electric motor powered by a battery. In addition, the plane will have solar cells on the main wings that will charge the battery. In order for this to be possible, the airframe needs to be able to handle this as well as any other features. First, the plane we use will likely need an overhead wing. This is opposed to a wing that goes under the fuselage or in the middle. The reason for the overhead wing is so that the plane will be able to glide easily without constant power from the motor. Also, planes with overhead wings are easier to fly which will be very important when attempting to fly the plane using only GPS guided coordinates. Since we will be using solar power to charge the battery, there is a good chance of losing power due to a lack of sunlight while in midair. An overhead wing greatly reduces the chances of our plane stalling and crashing due to loss of power from the motor.

Another important feature of the plane is the wing dimensions. The plane will require a wingspan between four to seven feet. The reason for having a wingspan of this size is to ensure that it will be able to carry all of the necessary features that will be added to it. In addition to the wingspan, the wing area needs to be at least 400 in². This amount of wing area is the minimum amount necessary to mount enough solar cells to adequately power the plane. Another important characteristic of our design is the planes ability to carry extra weight as deemed necessary. As we progress with this project we may decide to add additional features. These features will most likely require extra hardware to be attached to the plane which will increase the total weight. If the weight increases, we will need to scale the dimensions of the rest of the plane even higher to ensure that it is able to fly.

The motor of the plane is another significant part of our design. Obviously, the motor must be able to provide enough power to keep the plane in flight. Additionally, the motor must meet other specific needs. For example, the motor must be able to be powered by solar cells as well as the battery. Also, the motor should be able to quickly turn on and off during flight to conserve energy while gliding. If the original motor for the plane is unable to meet the demands of increased, we may need to replace it with a more powerful one. Another key aspect of the physical design of the plane is the fuselage. The fuselage needs to be large enough to hold all of the electronic parts. As the project continues, more features may be added requiring more hardware. However, if the fuselage becomes too large it will reduce the plane's ability to glide without power. The fuselage should be able to hold all of the parts while maintaining an aerodynamic form.

The aerodynamics of the plane will remain as a top priority. The main objective is to make sure our design is able to fly, so if any of the additional features interfere with this, changes will be necessary. For example, if the solar cells interfere with the airflow around the wings they may need to be laminated. This applies to every other part that is added to the frame of the plane as well. One of our goals is to be able to fly the plane

using only GPS guided instructions without a remote control. The plane will need to be steered solely from its rear flaps and rudders. This would be the simplest way to guide the plane without a remote control. In order for this to be possible the main wing will most likely need to remain stationary. A main wing without rudders is also beneficial by allowing the entire area of the wing to be mounted with solar cells.

3.2 Specifications

Listed below are the general specifications that are necessary to the success of the S.E.V. project. Each function has a range of values that outline the requirements for any airframe that may be well suited for the design. This allowed us to narrow down our search when looking for an appropriate plane that we could use as the physical frame.

Wing Area: at least 400 sq in
Wingspan: 4~7 ft
Length: 2.5~3.5 ft
Payload Capacity: at least 8.0 oz

These are the required specifications that the airframe must meet. The wing area is necessary so that we can fit enough solar cells. The solar cells each have dimensions of 10.6 in x 3.5 in. Combining the wingspan of 4~7 ft and an area of 400 sq in allows enough space for a minimum of 8 solar cells evenly spaced apart. The only requirement for the length is to be proportional to the wingspan for efficient flight. The payload capacity is one of the most significant requirements. Basically, the plane needs to be able to carry any additional devices that are attached to it and still fly.

3.3 Research

When the project was started, a great deal of time went into researching an appropriate frame for our Solar Extended Vehicle. We decided that the vehicle should be a plane which greatly limited the type of equipment that could be added to our vehicle. Additionally, this put weight limits on all of the extra electronics. Also, the amount of space that would be available was limited by this decision. With all of this background information in mind, we were able to narrow down our selection to the following airframes.

3.3.1 Apprentice 15e

This plane was recommended as a solid starting plane for beginners. Since the plan was to have an autonomous flying vehicle, it was an important factor for any of the potential planes to have a low amount of skill necessary to fly them. Another feature that was appealing about this particular plane was that it had a large wing area of 525 sq in which could accommodate large solar cells.

Pictured below in Figure 3.2.1 is the Apprentice 15E. The image shows the large wingspan that would provide a beneficial function in our S.E.V. design.



Figure 3.2.1 - Apprentice 15E
Reprinted with the permission of E-flite®

Specifications for the Apprentice 15E:

Wingspan: 58 in (1475 mm)

Wing Area: 525 sq in (33.7 sq dm)

Weight (w/o battery): 32-35 oz (910-1000 g)

Weight (w/ battery): 40-45 oz (1135-1275 g)

Motor: E-flite 15-size Brushless Outrunner

ESC: E-flite 30A Brushless ESC

Transmitter: Spektrum 2.4 GHz DX5e DSM2 5-Channel Air System

3.3.2 Parkzone® Radian

Another potential plane was the Parkzone® Radian. This also included several key aspects that we needed for our final design. This plane had the largest wingspan of 78.7 in. The curve in the wing would not pose a problem when fitting solar cells since we intended on using flexible cells that could conform to almost any smooth surface. Additionally, the propeller had the ability to fold while the motor is shut-off which would allow for gliding. The image below shows the Parkzone® Radian aircraft that was considered as a potential frame for our design.



Figure 3.2.2 - The Radian

Reprinted with the permission of Parkzone®

Specifications for the Parkzone® Radian:

Wingspan: 78.7 in (2.00 m)

Length: 44.6 in (1.14 m)

Weight (w/ battery): 30 oz (850 g)

Motor: 480-size 960 Kv brushless outrunner

ESC: E-flite Pro 30A brushless with switch-Mode BEC

Battery: 3S 11.1V 1300mAh Li-Po

3.3.3 The ProJet RQ-11

After considering the other models shown above, we decided to use the Raven B RQ-11. This plane included all of the ideal features that we were looking for. The first feature it had was a large wingspan that could fit large solar cells to produce efficient solar power. Second, the plane had enough space in the fuselage to accommodate all of the necessary electronics for the added features. Third, the static over-head wing allows the plane to glide easily while steered from the rear rudders without requiring precisely-controlled flying.

An image of the Raven B is presented below. While the ProJet RQ-11 is not an exact replica, Figure 3.3.1 shows a very accurate representation of what the final design will look like.



Figure 3.3.1 - RQ-11

Reprinted with permission of AeroVironment Inc (pending)

Specifications:

Endurance: 10-15 minutes (rechargeable battery)

Speed: 32-81 km/h (17-44 knots)

Operating Altitude: 100-500 ft (30-152 m)

Wingspan: 4.5 ft (1.4 m)

Length: 3.0 ft (0.9 m)

Weight: 4.2 lb (1.9 kg)

3.4 Functionality

Our Unmanned Aircraft System (UAS) will be expected to carry out numerous functions. The first and most important is flight. Our design will need to fly smoothly via both remote and autonomous navigation. Since the aircraft is predesigned to fly quite effectively, our goal is to limit the amount of extra weight that is added to the plane. A key component to our plane's flight capability is the overhead wing. This allows the plane to glide with minimal complications. In addition the majority of the steering is done from the rear tailfin. It is this feature that enables us to fly the aircraft autonomously without the precision that manned control would offer.

The plane is designed to take off by hand using an overhand throw technique. The lightweight nature of the vehicle makes this quite easy to firmly grip the plane with one hand and gently release the plane into the air to begin its ascent. To begin take-off, the aircraft is held by hand with the motor running at full power. Next, the plane is guided into the air in a smooth motion. During this process, the angle of take-off is very important. If the plane is tossed to steep, it will immediately stall and crash. In contrast, if the angle is too low, it will dive straight into the ground. Practice will be a key part of

executing this delicate process. There are a couple of techniques available to land the plane. One of them is known as a "deep-stall" landing. Basically, the aircraft descends until it is just a few feet above the ground. Once it is close enough the nose is steeply angled upwards to cause an intentional stall or "deep-stall". Once, the aircraft has successfully stalled it may either fall lightly to the ground or it can be caught by hand. Additionally, it may be possible to create an automated landing function for the plane. This would allow the plane to land autonomously using the same techniques as described previously.

Another method for take-off and landing the plane involves using landing gear. By using a set of wheels, a landing gear system could be used similar to that of a traditional plane. This technique would be beneficial by reducing the impact on the plane during the retrieval process. Due to this reduced impact, the plane would be capable of more flights before permanently damaging any of the parts. However, the addition of landing gear requires the total weight of the plane to be increased. Furthermore, in order to allow the landing gear to be retractable, more servos would need to be added for each wheel. Additional servos create more demand on the battery, which would result in a reduced total flight time. Also, these added servos put more of a load on the electronic speed controller and would require the use of more channels. While a landing gear system does provide some benefits, in the long run these benefits reduce the overall efficiency in weight and battery use.

Another key aspect of the S.E.V.'s functionality is its in-flight maneuverability. Our design differs from the majority of traditional planes in the way that it changes direction during flight. One feature that relates to this is the static main wing in contrast to a wing with hinged ailerons attached to each side. In most cases, a plane is maneuvered almost exclusively from the shifting direction of the ailerons while the rear rudders and elevator play a minor role by comparison. The S.E.V. does not contain ailerons, hence the static main wing. The function of the ailerons is replaced by several innovative components in our design. There are bends in the wing which allow the plane to make smooth turns during its flight path. Also, the Raven RQ-11B features a rear rudder and elevator similar to those included on a traditional plane. However, the role of the rudder and elevator is much more important to changing the orientation of the Raven compared to most other planes. The rudder and elevator are designed to work in conjunction with the curvature of the main wing to allow for flight maneuverability.

The Raven RQ-11B has several distinct features that separate it from other planes. The light weight and portable nature of the Raven makes it an ideal candidate for military operations. Furthermore, the ability to attach additional devices such as cameras and other sensors is another important benefit of this particular plane. The rear propeller is also notable because the placement of the motor in the center of the plane provides an ideal center of gravity. This allows us to add weight evenly around the plane without shifting the overall weight ratio and disrupting the aerodynamics. Another significant part of the airframe of the Raven is the overhead wing. An overhead wing is ideal for gliding with minimal power from the motor. Also, the bend in the main wing enables the plane to maneuver easily without the need for ailerons. The control of the pitch, roll, and turning

of the aircraft will be performed exclusively from the rear of the plane. Due to the simplicity of the plane's flying controls, the Raven can easily be adapted to fly completely autonomously. In addition to remote controlled flying, the design will also have the ability to fly autonomously via GPS. The absence of ailerons on the fixed-wing of the aircraft allows the Raven to be flown smoothly without the need for human-guided precision.

3.5 Motors

Once the airframe had been chosen, the next step was to find a suitable motor. The Raven RQ-11B can accommodate a varied a wide range of motors effectively. However, we need to select one that was the best fit for our particular design. The S.E.V. project has unique features that are new to the Raven frame. Due to these innovative changes, the motor will need to be chosen accordingly. Listed below are the potential motors that are best suited for this project.

3.5.1 Motor Research

The first choice for a suitable motor is the Alpha 480 Outrunner Brushless Motor. This motor is rated at 1020Kv and weighs 3.9 oz. This ratio is ideal for a plane weighing about 5 lbs.

The next possible motor is the Power 10 Brushless Outrunner Motor. This motor is slightly larger than the Alpha 480 and has a similar weight to Kv ratio. However, the idle current is almost twice as high. The choice of motor will depend on the battery that is selected for the Raven. The main appeal of this motor is the high RPM per volt in the Kv rating. This motor produces a high amount of power with a slightly higher weight.

Another potential motor is the ElectriFly Ammo 28-35-5100kV Brushless Electric Motor. Compared to the previous motors discussed above, this motor is the lightest in weight. Despite the light weight of the motor, the Kv output is 5100Kv and is much higher than the other motors that are being compared. However, 5100Kv is still a suitable rating for the Raven RQ-11B that will be used in our design. The lightweight nature of this motor allows us to maintain fewer loads on the plane while still providing plenty of power. The only disadvantage is the motor's power consumption which may drain the battery faster.

The Optima 480 Brushless Motor 1020Kv is another suitable motor that can be implemented into the design of the RQ-11. This motor is very similar to the 910Kv motor that was previously detailed since it is also part of the Optima series and they are both in the same size category. However, there is a noticeable difference in weight. The 1020Kv motor weighs 115g as opposed to the 910Kv motors which weighs 90g. Weight differences are always significant to our design since we are always trying to spare unnecessary weight additions. Another obvious difference is the increased Kv rating from 910Kv to 1020Kv. It is still yet to be determined whether the increased Kv rating will

increase the effectiveness of our design despite the added weight. The specifications for this motor are displayed below.

Model #	Alpha 480 (63M21)	EFLM4010A	Electrifly Ammo28-35- 5100kv	75M21- Optima480-3010- 1020kv
Size:	480-size	10-size	Optima480-size	Optima480-size
RPM / Volt:	1020 Kv	1100Kv	5100 Kv	1020 Kv
Max Voltage		7.2-12 V	7.4-11.1V	14.8 V
Input Watts:	275 W		333 W	275 W
Resistance:	0.06 ohms	.043 ohms	.057 ohms	.037 ohms
Idle Current:	1.10 A	2.10 A @ 10V		1.5 A
Continuous Current:	22 A	32 A	30 A	22 A
Max Burst Current:	28 A	42A (15 sec)	50 A	37 A
Cells Ni- Cd/Ni-MH:	6-10	6-20		
Cells Li-Po:	2-3	2-3	3	3
Weight:	3.9 oz (110 g)	122 g (4.3 oz)	94 g	115 g
Overall Diameter:	1.4 in (35 mm)	35mm (1.40 in)	28 mm	37 mm
Shaft Diameter:	0.16 in (4 mm)	5mm (.20 in)	3.2 mm	4 mm
Overall Length:	1.42 in (36 mm)	43mm (1.60 in)	35 mm	32 mm
Type:	Brushless out-runner	Brushless out- runner	Brushless in- runner	Brushless out- runner

Table 3.5.1 – Motor Spec Comparison

3.6 Servos

The Raven Rq-11B requires two servos in addition to the motor. While the motor is connected to the propeller, two servos are needed to control the rear rudder and elevator. The rudder and elevator is located on the tail end of the airframe. The rudder runs vertically with the tail fin while the elevator is connected on a horizontal hinge below the rudder. These two flaps will be the main method of maneuvering the plane while it is controlled via remote control or by GPS navigation.

3.6.1 Servo Research

The EXI S1123 9G Micro Servo is the servo that will be used in the SEV design. The design requires two identical servos of this same type that will steer the rear rudder and elevator. Generally, servos are not known for their reliability. This means that they are more than likely to be easily damaged in test launches and will need to be replaced. The low cost of \$2.88 makes these parts very cost effective for this project.

Another possible servo is the T-Pro MG90S. This particular servo is better overall in quality which results in an increase in cost. However, it is yet to be determined whether or not the servo shown below is more cost-effective than the previous one.

The E-Sky 8g High Speed Mini Servo is a fast, digital servo that is suitable for our design. The exact weight is 7.5g which makes it slightly less than both of the other potential servos. Additionally, the operating speed performance is not decreased by the reduction in size as is usually the case for smaller dimensions. Also, the cost is about equal to the price of the T-Pro MG90S. The specifications for the described servo are listed below.

Model	EXI S1123 9G Micro Servo	T-Pro MG90S	E-Sky 8g High Speed
Dimensions	22.6 x 11.4 x 22.2mm	23 x 12.2 x 29mm	22.8 * 11.5 * 20.8mm
Weight	9g (0.32oz)	9g (0.32oz)	7.5g
Operating Speed (4.8V)	0.12sec/60 degrees	0.11sec/60degree	60 degrees / 0.1s
Torque (4.8V)	1.4 kg/cm	2.0kg/cm	> 1KG/cm
Operating Voltage	4.8 V	4.8-6.0V	DC5V +/- 1V

Table 3.6.1 – Servo Spec Comparison

3.7 Electronic Speed Controllers (ESC)

The Electronic Speed Controller or ESC is the third major electronic component that is necessary to allow the plane to fly effectively. The ESC will work in sync with the motor to provide smooth acceleration during flight. Basically, the ESC is what regulates the throttle to the motor. This is a significant role for the electronics portion of our design, so we needed to research accordingly. The Selection of the speed controller would be very dependent on the specifications of the motor that is used. Since there are multiple choices for applicable motors, we needed to research appropriate speed controllers that would work efficiently with each motor.

3.7.1 ESC Research:

When selecting a speed controller it is important to find one that compliments the motor. The E-flite 60-Amp Lite Pro SB brushless ESC is a suitable match for the Power 10

Brushless Outrunner Motor. This speed controller allows for a continuous max current of 60A which is high enough for the motor's 32A current. Similarly, the voltage rating is also well-suited to be paired with the 10-size motor.

Another ESC that should be noted for consideration is the Volcano 40A Brushless Electric Motor Speed Controller ESC. This speed controller is compatible with the Alpha 480 Outrunner Brushless Motor. Additionally, the ESC will also work effectively with the servos that were listed above in the previous section. The current and voltage specifications have been matched appropriately.

The next speed controller is the Volcano Series 60A Brushless Speed Controller ESC. One of the significant attributes about this speed controller in particular is that it has a continuous current rating of 60A. This enables us to use a higher rated motor if it is found necessary for the project design. This speed controller is also larger in size in comparison to the other speed controllers that are being considered.

Another suitable electronic speed controller is the Platinum-40A-PRO Brushless ESC V1. This ESC provides a steady 40A current output which is an appropriate match for our selection of motors. Additionally the low weight of 38g makes this ESC the 2nd lightest ESC of our choices. The specifications for this ESC are shown below:

Model	E-flite 40-Amp Lite Pro SB brushless	Volcano-40 (07E05)	Volcano-60 (07E06)	Platinum-40A-PRO Brushless ESC V1
Cont. Current	60A	40A	60A	40A
Burst Current	75A	55A	80A	60A
BEC Output	5V/2.5A	5V/3A	5V/3A	5V/3A
Input Voltage	3-6 cells (LiPo)	2-5 cells (LiPo)	2-6 cells (LiPo)	2-6 cells (LiPo)
Weight	66g	33g	60g	38g
Dimensions	33x76x13mm	55x28x12mm	70x31x14mm	59x 27x12mm

Table 3.7.1 – ESC Spec Comparison

3.8 Integration

This section describes how the core airframe electronics communicate with each other in order to perform independent functions simultaneously. The plane is based on a 3-channel system which integrates the motor and two servos. The motor will be the primary source of power for the SEV while the two servos' main functions will control the rear rudder and elevator. The electronic speed controller serves as a connection between the remote control and the motor as well as the servos. The main function of the ESC is to

work as a throttle to make precise changes in the plane's airspeed. Furthermore, the ESC also dictates the action of the servos. The speed controller gives command to the servos which thereby translates to the rudder and elevator. This communication allows the user to maneuver the plane by shifting the alignment of the rudder and elevator. Since the plane does not have ailerons on the main wing, the orientation of the rear rudder and elevator will be the primary method of turning the plane during flight. To summarize, the remote control communicates with the electronic speed controller which, in turn, sends instructions to the motor and the servos. Using this system, the user is given full control of both the speed of the plane as well as the pitch and roll to change directions during flight.

The following diagram shows the series of connections between the motor, servos, and ESC components relative to the user control. This is the most basic system that is required for a radio-controlled aircraft.

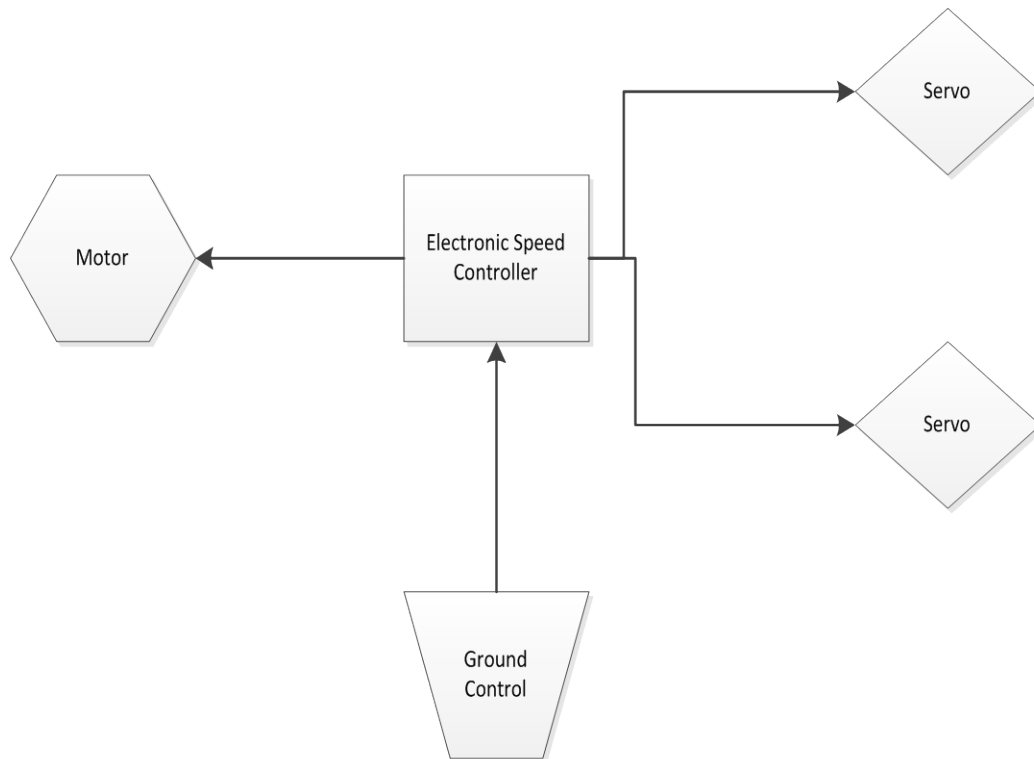


Figure 3.8.1 – ESC Functions

Chapter 4: Printed Circuit Boards

4.1 Circuit Board Requirements

For this project and senior design we will be producing a printed circuit board or PCB. The plan is to have two PCBs; one PCB will have the autopilot circuitry and connectors

for the GPS, Xbee, and servos. The second PCB will be our power board. This board will contain the circuitry for charging the battery. We have decided to do this because; this allows us to keep the power on its own circuit as a failsafe. We would rather burnout one board than both boards. With the charging circuit on its own board, this also allows us to keep the temperatures down. The sizes of the PCBs are also very important. The airframe is limited on space; therefore, the PCBs will also need to be as compact as possible, while allowing good airflow through the boards to prevent overheating.

PCBs consist of two main materials that can vary depending of the application of use. These materials are the conducting layers and the insulating dielectric layers. The most common dielectric used in printed circuit board fabrication is FR-4, woven glass reinforced sheet of epoxy. PCBs also come in multiple layers; for this project we will need no more than two layers. Along with the layers, the PCBs can also come in many different colors. These colors come from the solder mask that is paced on the board where solder is undesired. This has no effect on the project, but school colors would be nice. PCBs will need to support surface mounted devices and with a few through-hole connectors. Using surface mounted devices allows us to save space in height and width. With saving space in height this will allow to stack the two PCBs on top of one another. So, when we are choosing the microcontrollers and sensors, we will be using a TQFP package type. Below in Figure 4.1.1 is an ATmega328 TQFP microcontroller.

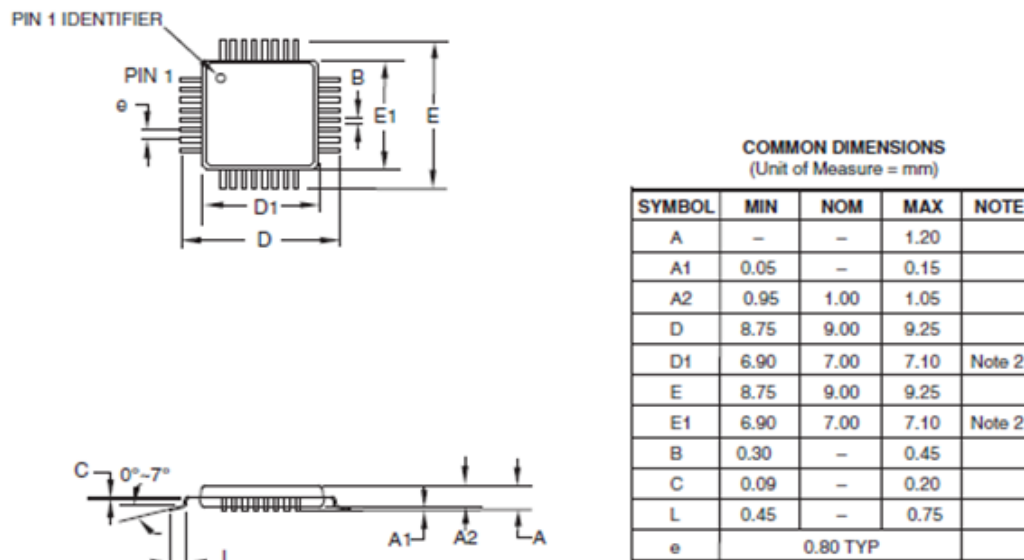


Figure 4.1.1 – ATmega328
(reprinted with permission of Atmel-pending)

4.2 PCB Vendors

After knowing what we are looking for in a PCB, we now need to find a vendor to produce the board. There are many vendors in the state of Florida and also in the world. This is important because of possible time constraints; we need to allow enough time to build the boards and test. With this statement, brings me to another point; some vendors offer assembly of the boards. This could be quite helpful, but could add to the budget. All

these vendors also offer many options for the boards too. These options range from multiple layers to solder mask color. One thing to we need to watch out for is the file type excepted by the vendors. What is meant here is which program is used to create the schematic of the PCB. So, when choosing which vendor to use we need to confirm if we have access to the schematic software.

Below in Table 4.2.1 we have a list of specification from different vendors. If the section is blank, that information was unable to be obtained. The cost of the board will vary by size. Plus, most vendors either sell the boards as a set of 5 or a set of 10. For this project limiting the amount of boards will help the budget; so we will be looking for smaller sets.

While researching PCBs for our design, we encountered several different vendors with PCB specializations. We determined the best way to find the most suitable PCB was to create a side-by-side comparison.

Table 4.2.1 – Comparison of features between different PCB vendors.

Company:	<u>Circuitronics Corp</u>	<u>AEM</u>	<u>JetPCB</u>
<i>Specifications:</i>			
Layers	2 to 20	2-18	1-20
Maximum Board Size		24"x36"	19"x13"
Board Thickness	0.063"(others optional)	4L:0.016", 6L:0.024", 8L:0.032", 10L:0.048"	2L:0.012"(others optional)
Copper Weight	1oz	0.5-4oz	1-4oz
Minimum Hole Size	13mil	8mil	2mil
Trace & Space:	6mil, 6mil	5mil, 5mil	
Maximum Hole Size	250mil		250mil
Silk Screen Layers	2		
Solder Mask Color	green		G, LG, MG, W, B, Y, R, BL
Un-plated Holes?			Yes
Notches & Cutouts?			Yes
Irregular Shape?			Yes
Testing?	yes	Yes	Yes
File Types			Gerber274x,d, DPF, ODB++
Cost:	Contact for price quote	Contact for price quote	Contact for price quote

Source: All information in the table was obtained through vendor websites.

Company:	<u>DorkBot</u>	<u>ILead Studios</u>	<u>PCB Fab Express</u>
<i>Specifications:</i>			
Layers	2 or 4	2	2,4,6
Maximum Board Size		5cmx5cm or 10cmx10cm	
Board Thickness	.062"	.063"(others optional)	.062"
Copper Weight	1oz	1oz	1oz
Minimum Hole Size	13mil	12mil	15mil
Trace & Space:	6mil, 6mil	6mil, 6mil	6mil, 6mil
Maximum Hole Size	250mil	250mil	246mil
Silk Screen Layers	2	2	1
Solder Mask Color	purple	Blue, white, yellow, red	Green
Un-plated Holes?			Extra charge
Notches & Cutouts?			no
Irregular Shape?			
Testing?		yes	Extra charge
File Types	Gerber274x, Eagle.brd	Gerber274x	Gerber274x
Cost:	\$5/sq.in. for 3 copies	5x5-\$25 for 10copies, 10x10- \$38 for 10 copies	5 day turnaround 5boards: 2layers- \$15/board

4.3 PCB Specifications

Now we have a selection of vendors to choose from, plus we also know what each one offers; so, we can begin to design our board. For our project, we will be designing two boards; below is a list of specifications for each of our project's PCBs.

Autopilot PCB

- 2 layers
- Length <10cm
- Width <5cm
- Surface Mount Technology
- Black solder mask (optional)
- Yellow silk screen (optional)
- Through Hole supported
- Board weight without components <1ounce
- Board weight with components < 2ounce

Power PCB

- 2 layers
- Length <5cm
- Width <5cm
- Surface Mount Technology
- Black solder mask (optional)
- Yellow silk screen (optional)
- Through Hole supported
- Board weight without components <.5ounces
- Board weight with components < 1ounce

4.4 Board Layout

The following section shows the desired layout of the PCBs. When the final Eagle files are complete the PCBs should be of the same design as the ones in Figures 4.4.1 and Figure 4.4.2. Below in Figure 4.4.1 you can see we are going to have a narrow, but long, PCB at 9cm by 3cm. This is because of the space limitation in the fuselage of the UAV. Starting from the right, you notice the three sensors: X/Y gyro, Z gyro, and the Accelerometer are located near the right edge; the reasoning for this placement is when the board is inserted into the UAV those sensors will be closer to the center of gravity. This will allow for more precise readings of the UAV's orientation in space. This also lets us know that this is the rear end of the PCB. The white arrow at the left side of the PCB also denotes direction of placement; this is for gaining the correct bearing.

If you bring your attention to the top right and bottom right sections of the PCB you will notice two connectors. These connectors are gold pin connectors, as of now, are only used to hold the power PCB above itself. These connectors could be used to create a power connection if needed. You will also notice that those connectors are not the full length of the PCB this is because the power PCB will be half the size of this board. Plus, on the left half of the PCB are the connectors for the GPS, RC Rx and the servos. If the power PCB were the same length it would interfere with these connections.

The left half side of the PCB is the navigation unit, which the data is being collected by all the collective devices such as the gyroscopes, the accelerometer, the GPS, and the barometric pressure sensor if needed in the future. All the collective devices send their data to the microcontroller on the navigation unit. Next, the Autopilot microcontroller accepts all the collected data and responds accordingly by shutting off the motor if the SEV at a very high altitude and controlling the servos to stabilize the SEV. All the data are sent to the ground station via Xbee which is to be connected to the autopilot microcontroller.

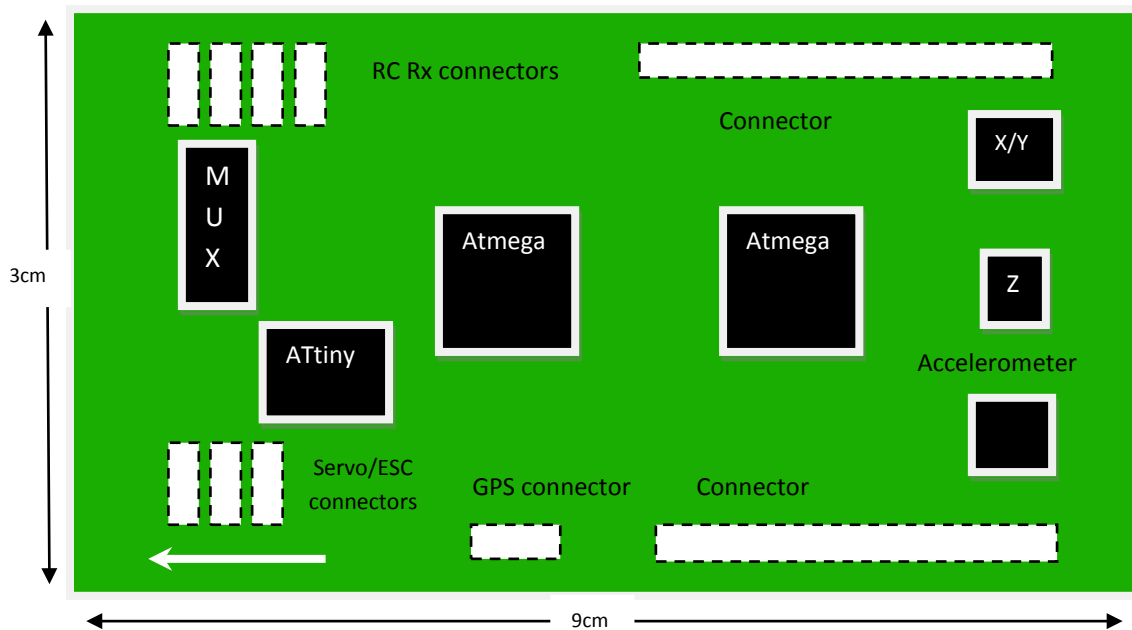


Figure 4.4.1 – Layout of the Autopilot Board.

Below in Figure 4.4.2 is the PCB layout of the charging circuit. This board is half the length of the autopilot PCB at 4.5cm by 3cm; these two boards share the same width. This allows the board to be stacked and fitted into the fuselage. For this stacking to work, you will notice the two dotted-line boxes, in Figure 4.4.2, at the top and bottom of the figure. These connectors are on the bottom side of the PCB. These connectors will receive the gold pin connectors on the autopilot PCB. This type of design should help with keeping the temperature in the fuselage down. It will allow air to pass over the top and bottom of the charging PCB. On the top side of the PCB we have three connectors: one for the battery, the Solar cells, and the electronic speed controller (ESC).

On a regular setup of an RC aircraft the battery is connected directly to the ESC. For this project the battery will be connected to the charging circuit. This allows us to charge the battery during flight. Then the ESC will be connected to the battery through the board. The ESC will then need to be connected to the autopilot PCB, because this how the board is powered. The solar panels are connected to the maximum power point tracker, LT3652. The maximum power point tracker actually tracks the output current at a specific voltage level. Once it finds the maximum current flow, it captures the current to charge the lithium-polymer battery. The battery is going to be charged and discharged at continuously by connecting the battery to both the charging circuit and the ESC to power the motor and the autopilot board.

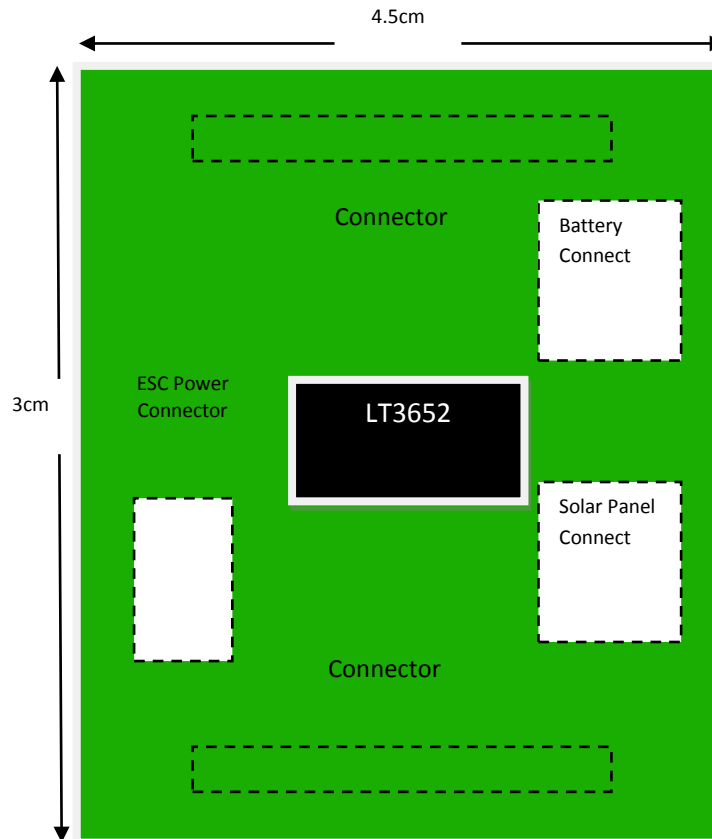


Figure 4.4.2-The Charging Circuit PCB

4.5 PCB Testing

Before we install the boards into our UAV, we will need to test them to confirm that they are working properly. First we will do a visual inspection of all the parts solder to the board; whether they have been solder by us or a vendor. After confirming the build quality, then the software will be loaded to the board. The board then will be tested using simulator software to verify that the code is working correctly with the new autopilot PCB. Meanwhile, the power PCB will be connected to the solar cells to verify that the battery is charging from the cells. Then the two PCBs will be connected together to confirm that the power PCB will power the autopilot board. Then the two boards will be tested again using the computer simulator. Once we believe that the two boards are working correctly, we will then install them into the UAV for flight testing.

4.6 PCB Installation

After the simulator testing, the boards will need to be installed into the UAV. The PCBs will reside in the fuselage of the aircraft. Below in Figure 4.6.1 is an interior photograph of the UAV.

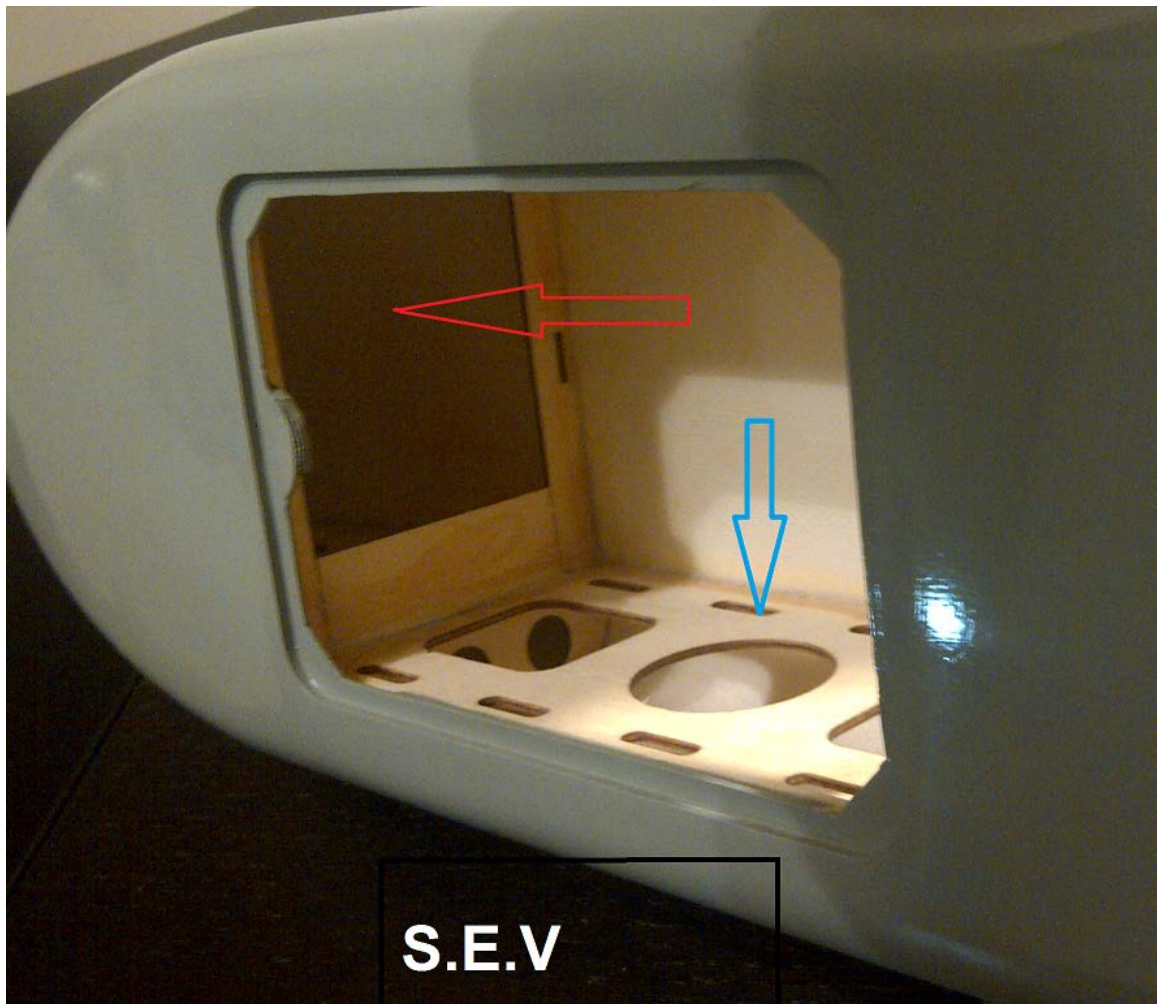


Figure 4.6.1 – Inside the fuselage of the UAV

In Figure 4.6.1 you can see two arrows pointing towards two different locations inside of the UAV. The red arrow is point to the nose section of the fuselage. This area is small; the PCBs could fit into this area, but we need to have the sensors closer to the aircraft's center of gravity. This area can be used to place a camera instead. The blue arrow is pointing to the area where we would like to place the PCBs. This area will also be shared with the battery, Xbee module, and the RC Rx. The RC Rx can be moved into the nose of the aircraft and the Xbee could be mounted to the top, to allow the antenna to stick out. The battery will need to be secured to the bottom with Velcro, to keep it from moving around during the flight. This leaves us with only one place to put the PCBs; that would be right above the battery. To do this we will build a mount that will rest on top of the battery, while letting the battery hold the mount secure. Below in Figure 4.6.2 is a rough sketch of that mounting system.

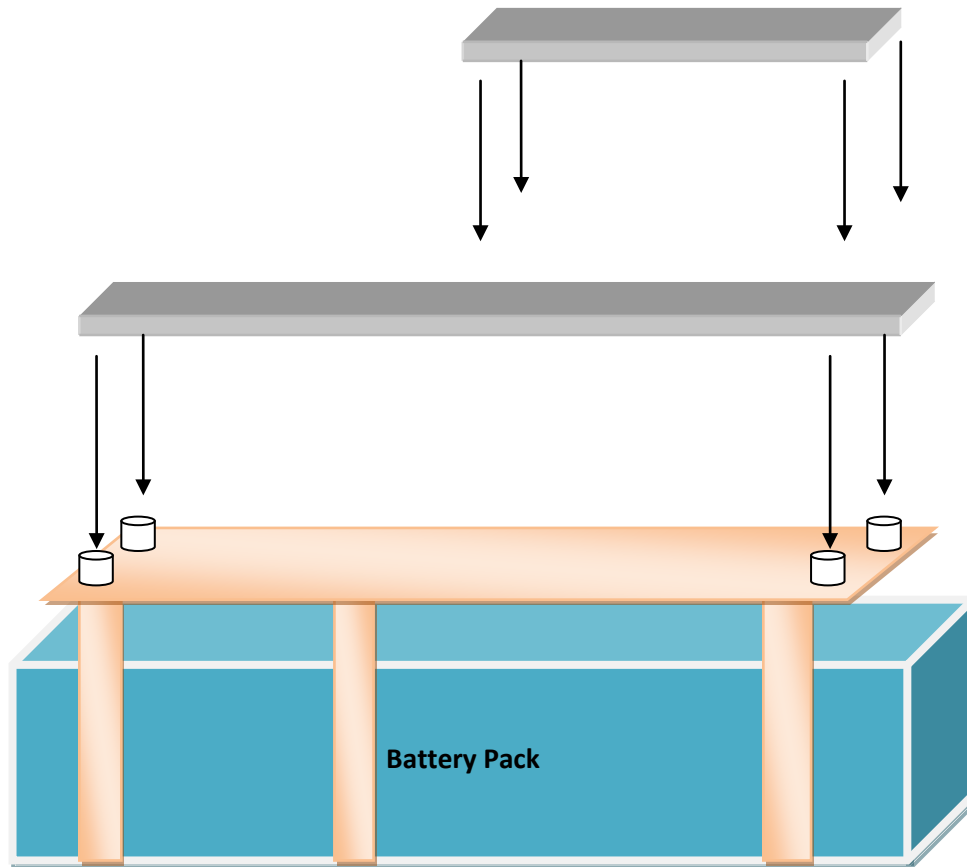


Figure 4.6.2 – The battery back pack mounting of the PCBs

In Figure 4.6.2, the two dark gray pieces are the PCBs; again these are connected through the gold pin connectors that will be placed on them. These connectors are not represented in this diagram. The two PCBs will then rest on the four white spacers to keep the board of the mount. The mount will be constructed with balsa wood and wood glue. Balsa wood is very light weight, thus making it a good material to use during this project. The mount will not rest directly against the battery; it will be raised up allowing the battery to cool during operation. This mount will be connected to the battery by Velcro. Then attached to the bottom of the battery are two strips of Velcro, which will attach the whole mounting system to the fuselage. The battery's weight plus the Velcro will keep the PCBs stable and level; which is important when calibrating the sensors.

Chapter 5: Microcontroller

5.1 Microcontroller Requirements

There are many microcontrollers in the world today, so choosing which controller to use can be a difficult task. For this project we will ease this decision by choosing the

microcontroller by what software we can use to program it. With this we can begin to narrow down which device to use. The best option is to go with an open source Integrated Development Environment (IDE) for one very important reason, it's free. For this project the Arduino IDE has been chosen, therefore the field of microcontrollers has been narrow down to Arduino compatible microcontrollers. Now we will look at what the most common features that are offered; such as: clock frequency, package type, number of analog to digital converters (ADC), number of Input/Outputs, and program memory type. The spacing for this project is a very crucial; the fuselage only has limited room to accommodate the battery, Xbee, RC Rx, and the PCB that will controller the UAV. Therefore the PCB will need to be as small as possible. With the microcontrollers coming in various packages; package type will also bear significant weight as it will determine how much real estate on our board. So for this project the TQFP package will be chosen for the microcontroller. It should be noted that choosing the TQFP will be difficult to solder. As much as space is a concern, so is the power the chip requires. Much of this project will be running on the power of a battery or solar cell. Much of the hardware does not need more than six volts, excluding the motor and servos which require a higher voltage. The microcontroller should be low powered and still achieve a high clock cycle to handle the sensors and real time processing.

The microcontroller needs to have the capacity to handle multiple inputs from various sensors. This means that it must support analog to digital conversion (ADC). One of the microcontrollers will be monitoring 2 gyroscopes and an accelerometer. So we will need at least need 4 ADCs. The microcontroller will also be connected to servos, so it will need to support pulse width modulation (PWM). Along with be connected to other digital devices; the microcontrollers will need to be connected to each other. This means that they will need to support a byte oriented 2-wire serial interface. For this project the UAV will be flying to set waypoints. These waypoints will need to be stored on non-volatile memory; so that if we have a power loss, we will not lose the waypoints. There many types of memory to choose from and many microcontrollers support multiple formats. We will be using an Electrically Erasable Programmable Read Only Memory Device (EEPROM) as it will prove much easier to work with during testing. The temperature is a small concern, because we will have all these electrical devices in the same small fuselage. Most of the testing will occur during the fall semester, so the outside temperature will be much cooler. This will help with keeping the temperatures down.

5.2 Comparison and Selection

5.2.1 ATmega1280

The ATmega1280 is a high-performance, low-power, 8-bit AVR RISC-based microcontroller. It is a combination of 128KB ISP flash memory, 8KB SRAM, and 4 KB EEPROM. The microcontroller has 86 general purpose I/O lines, 32 general purpose working registers, a real time counter, and six flexible timer/counters with compare modes. It also has PWM, 4 USARTs, 16-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. The device achieves a throughput of 16 MIPS at 16

MHz and operates between 2.7-5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves a through put approaching 1 MIPS per MHz, balancing power consumption and processing speed. [1]

5.2.2 ATmega328

The ATmega328 is a high-performance 8-bit AVR RISC-based microcontroller that utilizes picoPower®. The microcontroller uses combination of 16KB ISP flash memory with read-while-write capabilities, 1KB EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, SPI serial port, a 8-channel 10-bit A/D converter (only in the TQFP package), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. The ATmega328 has a maximum operating frequency of 20 MHz. By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed. [2]

5.2.3 ATtiny45

The ATtiny45 is a high-performance, low-power Atmel 8-bit AVR RISC-based microcontroller. It is a combination of 4KB ISP flash memory, 256-Byte EEPROM, and 256B SRAM. The microcontroller has 6 general purpose I/O lines, 32 general purpose working registers, one 8-bit timer/counter with compare modes, one 8-bit high speed timer/counter, USI, internal and external Interrupts. It has a 4-channel 10-bit A/D converter and a programmable watchdog timer with internal oscillator. There are three software selectable power saving modes, and debug wire for on-chip debugging. The device achieves a throughput of 20 MIPS at 20 MHz and operates between 2.7-5.5 volts. [3]

The ATtiny45 will be used for failsafe purpose only. It will be used in conjunction with a multiplexer. This failsafe uses a separate circuit to transfer control from the RC system to the autopilot and back again. This also allows for mid-flight rebooting of the main microcontroller. So for the project we will be using an ATtiny45. Based on the brief summary above, it can be concluded that the ATmega1280 is a much more robust microcontroller than the ATmega328. The memory on the ATmega1280 is exceeding any minimum requirements set forth. But this also creates waste, if not all of the memory is going to be used during programming. The ATmega328 poses a problem with this same feature just on the reverse end. If the programming for the ATmega328 becomes too large memory can become scarce and could cause operating errors. So to solve this problem, the use of multiple microcontrollers could be implemented. Whereas, the ATmega1280 could be the only microcontroller being used, thus cutting down on space.

The ATmega1280 also uses a bit more power when running the chip in low power. Some of the features of the low power for the ATmega1280 include: use of hardware DMA and

event system to offload the CPU, cut clock or supply on device portions not in use, careful balance of high performance and low leakage transistors, fast wake up from low power modes, and low voltage operation. On the other hand, the ATmega328 takes these features a bit further. Under the description of the ATmega328; the term **picoPower®** is used to describe the microcontroller's ability to function with a minimum of 1.67V while still maintaining all functionality, including analog functions. To achieve this, the device implements the use of sleep modes and flexible clocking options, plus the benefit of better transistor design and process geometry.

Below in Figures 5.2.1 and below in Figure 5.2.2, the TQFP packages have been chosen; there are two other types of package design for surface mounting, but the TQFP has been chosen due to the accessibility of equipment and the ease of securing the microcontrollers to the PCBs. In Figure 5.2.1 is a representation of the footprint that the ATmega1280 would have been placed on the printed circuit board (PCB). If you draw attention to the 'E & D', you can see that the chip is 16mm by 16mm.

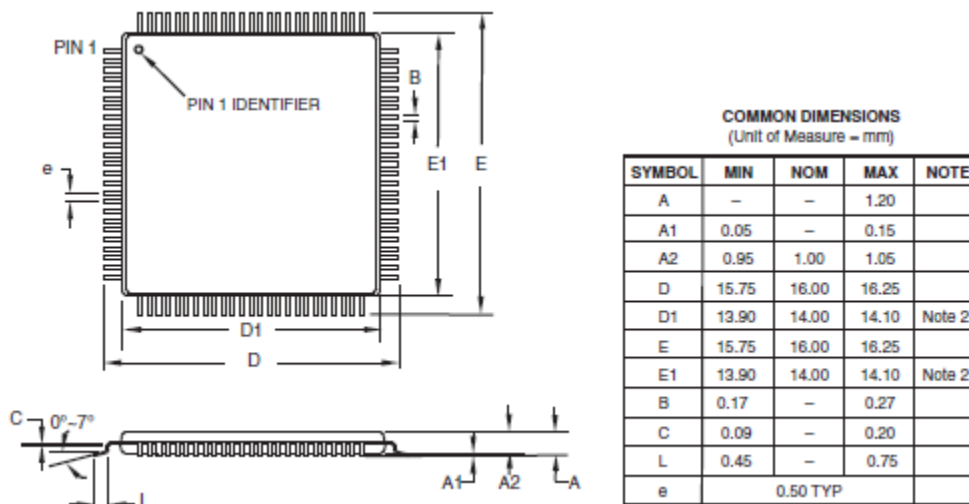


Figure 5.2.1 - ATmega1280

Reprinted with permission of Atmel

Below in Figure 5.2.2, the ATmega328 is about half the size of the ATmega1280. The measurements of 'E & D' are 9mm by 9mm, giving this microcontroller a considerably smaller footprint. Because of this, and the ability to run at the same clock speeds plus the use of **picoPower®** make this microcontroller an ideal choice for this project. For this project two of the shown below microcontrollers are to be installed to operate the system on two simple codes instead of one compound code.

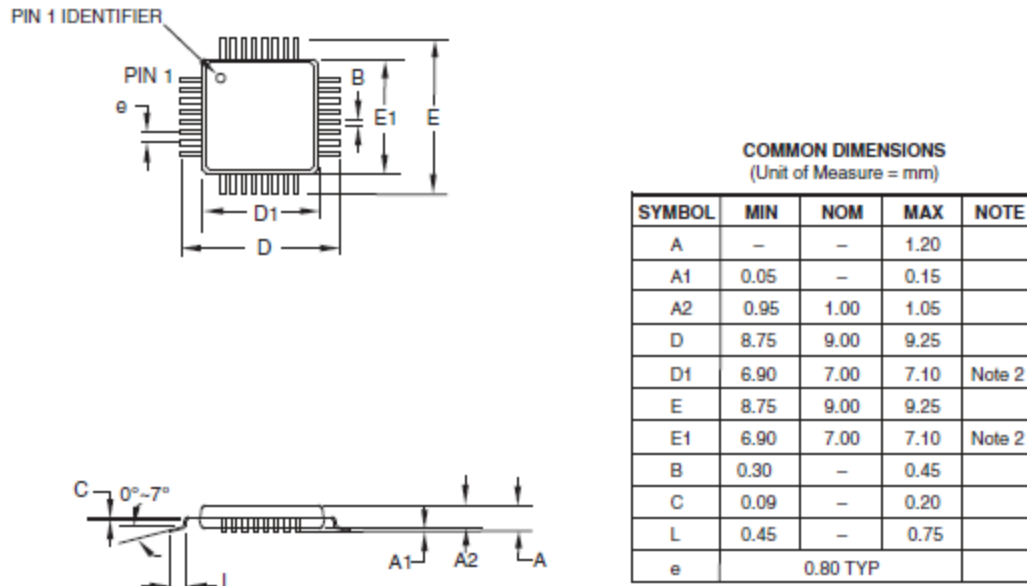


Figure 5.2.2 - ATmega328
Reprinted with permission of Atmel

Pricing is another factor to be considered when choosing which microcontroller to go with. The plan is to contact Atmel and request samples of the chips to help keep the project on budget. The current price for a single Atmega328 is currently listed at \$4.25 on the Sparkfun website. DigiKey has the same part listed for \$3.83, so depending whether or not Atmel approves the request for samples, DigiKey might be the preferred supplier for the components. The Atmega1280 is listed for \$16.13 on DigiKey; therefore selecting the ATmega328 will be ideal even if more than one microcontroller is needed.

These two microcontrollers have been chosen to be compared because they are similar projects that have been proven to work with the program that will be used for this project. There is one more microcontroller which is a newer version of the Atmega1280, which is the Atmega2560. The Atmega2560 is very similar in specifications to the Atmega1280, but with the program memory size doubled. The cost of this chip is also higher at about \$18.00 on DigiKey.

5.3 Microcontroller Integration

After selecting the ATmega328, the next step is how to integrate the microcontroller into the project. For this project two microcontrollers will be used; one chip will be used for the controlling the servos and motor therefore this microcontroller will be referred to as the Pilot controller. The Pilot controller will also monitor the charge on the battery and determine if it is time for the UVA to return to the launching point. The second microcontroller will be used for the navigation using the Altitude and Heading Reference System (AHRS) code, which is based on Bill Premerlani's Directional Cosine Matrix (DCM) algorithm. The second controller will be referred to as the Navigator.

The Pilot controller will, as the name denotes, fly the UAV. It will execute a series of instructions that are hardcoded onto the microcontroller to adjust the servos to correct its bearing and altitude during flight. The servos will be connected to the rudder and the elevator. The Pilot controller will receive the GPS position from the Navigator and will correct the servo position to correct the path of flight by either raising the elevator to increase the altitude or moving the rudder left or right to correct the UAV's bearing. The Pilot controller will also be checking the battery voltage and if the voltage on the battery reaches 15%; the Pilot controller will access the additional ram on the PCB and retrieve the starting coordinates of the UAV and then correct the course to return the vehicle safely.

Below in Figure 5.3.1 is the diagram of the communication between the devices and the Pilot controller. Starting from the left, the RC Rx has three channels (the two servos and motor) when the user is flying the UAV manually the Pilot controller will assist to provide a more stable flight. Next the Navigator will communicate to the Pilot controller sending the current position and adjustments that are required to correct the path or the altitude of the UAV. Then the Pilot controller will transmit a message to the Navigator requesting the next position and altitude check.

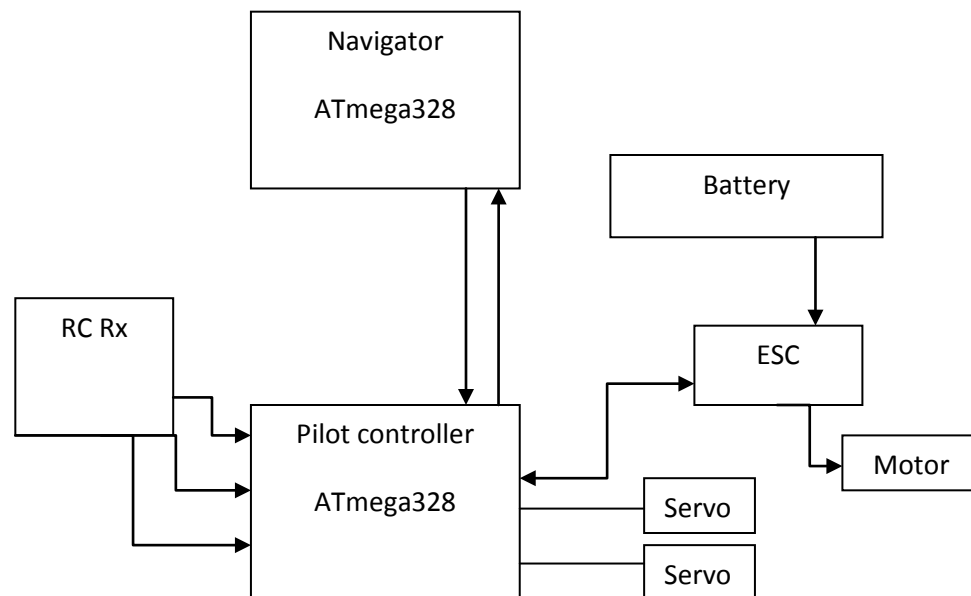


Figure - 5.3.1 - Autopilot Unit

During this communication the some bits of information will be written to the EEPROM. The data that will be written is the home location of the take-off and will also contain the location of the previous waypoint. Not shown on the diagram, there is a hardware connection between the battery and the Pilot controller that will allow the Pilot controller to monitor the voltage to determine if the UVA needs to return to the take-off location; in other words home. Next, you can see in Figure 5.3.1 that the Pilot controller is connected

to the electronic speed controller (ESC). This allows the Pilot controller to adjust the speed of the UAV.

The design goal here is to not run the motor constantly; we want the UAV to glide between a set maximum and minimum altitudes to prolong the battery life. To do this, the Navigator will receive the altitude from the GPS, perform an altitude correction, this is because the altitude is taken at sea level so it needs to be offset to get an accurate altitude of the UAV. Then that altitude is transmitted to the Pilot controller; where then it can begin to adjust the elevator to increase the altitude or level out the UAV. The pilot controller will both turn the motor off through ESC and allow the UAV to glide to the minimum altitude or turn the motor to 70-80% output and allow the UAV to climb to the maximum altitude. All the UAV's movements are due to the servos. The servos are connect to the Pilot controller through pulse width modulation (PWM), thus allowing the Pilot controller to know and change the servo's position. In Figure 5.3.2 is the diagram of the communication between the Navigator and its supporting devices. Beginning from the top left you have the X&Y gyroscope; this device will be used to determine the pitch (Y axis) and roll (X axis),the next sensor below the X&Y gyro is the Z gyro. This sensor is for measuring yaw (Z axis) or left and right orientation.

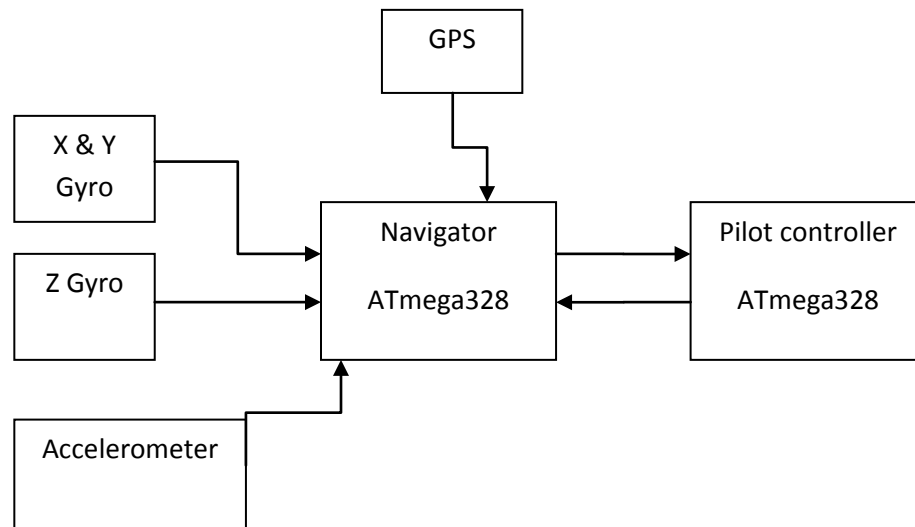


Figure 5.3.2 – Navigation Unit

With these two sensors the orientation of the UAV can be determined in a three dimensional plane; this information is then transmitted to the Navigator that is running the AHRS code. The accelerometer will transmit any directional movement of the UAV to the Navigator. This information is then used to calculate the correct adjustments needed to keep the UAV on track to the next way point or back to the home location. This information, as depicted in Figure 5.3.2, is then transmitted to the Pilot controller where it then will make the flight adjustments.

Chapter 6: Sensors: Roll, Pitch and Yaw

6.1 Sensors

The word “sense” when it is used as a verb it indicates awareness of something that is achieved by some type of sense organs. As humans have their five senses which enable them to be aware of their surroundings, the SEV is also required to be aware of its surroundings to do its tasks successfully. The sensors that are to be installed in the SEV are mainly used to help in flying, landing, and stabilizing the SEV while it is gliding by sensing or reading the physical quantities which are associated with the required functions. Since the SEV could be used in various fields, its tasks vary depending on which field it is being used for. Therefore, the initial design for the SEV is to operate very well such as functioning how to fly for a reasonable period of time and how to land safely which later allows for more integration to serve the best interest of the consumer. Therefore, for the sake of achieving the initial design, the SEV basically needs at least three sensors: gyro, accelerometer, and barometric pressure sensor.

6.2 Gyro

The angular rate sensors differ in the number of axes which are being angularly sensed. Triple axes gyros are used to sense the angular displacement in degrees with respect to the x-axis, y-axis, and z-axis. For better understanding of the axes, they will be called longitudinal, lateral, and vertical respectively. The triple axes gyros sense the angular displacement with respect to the longitudinal axis due to the rolling of the SEV. In addition, it senses the angular displacement with respect to the lateral axis due to the elevator movement of the SEV; finally, it senses the angular displacement with respect to the vertical axis due to the rudder movement of the SEV. Figure 6.1 shows and explains how the SEV rolls, pitches and yaws graphically.

The gyroscope is a crucial device on the circuit board; it is needed to report the angular position of the SEV with respect to the three axes. However, the decision on either a triple axis gyroscope or a dual axis gyroscope combined with a single axis gyroscope has been left to the software designer. Coding a triple axis gyroscope might be unpleasant which gives an advantage to the double axis gyroscope combined with a single axis gyroscope. At the end, sensing the angular position of the SEV will allow the servos to adjust themselves to keep the SEV stable.

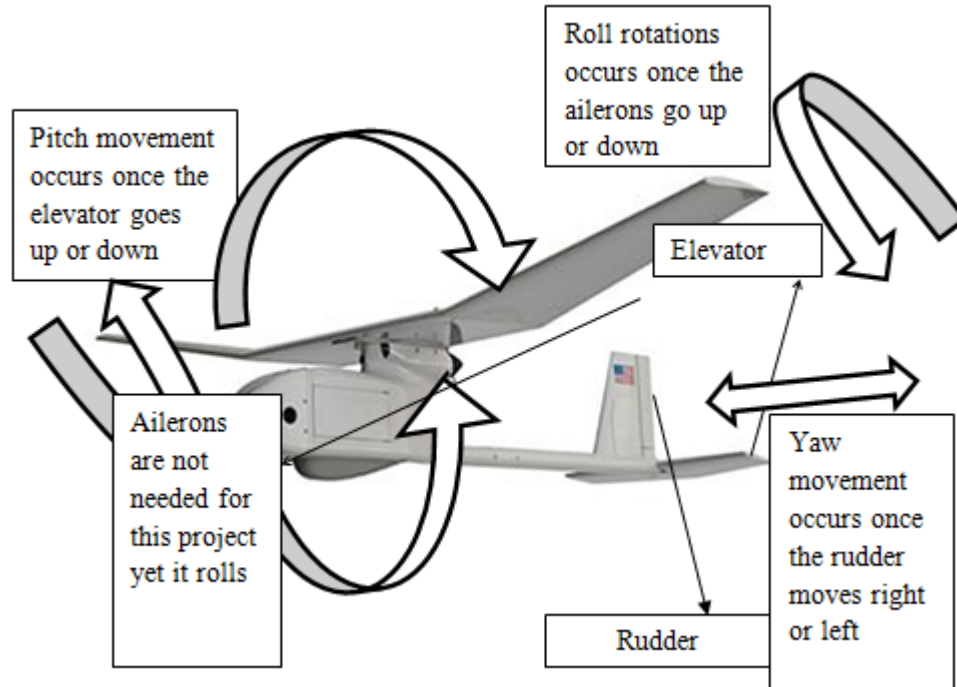


Figure 6.1 - Roll, Pitch, and Yaw.

Reprinted with the permission of AeroVironment -pending

6.2.1 Basic Requirements

The market is diverse regarding finding a specific gyro. Therefore, some requirements are set regarding the need of either one axis or dual axis or triple axis gyroscope. In addition to the other factors such as power consumption, accuracy, size, and weight, Table 6.1 represents the general requirements to reduce the search as much as possible to save time and to come up with what serves the needs of the project.

Gyro	General Requirements
Number of Axis	Since the plane is predesigned to be without ailerons, a dual axis gyro is preferred to measure the pitch and roll of the SEV as well as a single axis gyro to measure the yaw movement of the SEV
Power	Since the SEV is supplied with a limited source of power, the minimum power consumption is preferred
Sensitivity	Since the SEV is going to be in the sky, high sensitivity is preferred to track how the SEV behaves with the winds
Weight	The lighter it is; the better it serves the overall design, and more specifically it eliminates the power to weight ratio issue
Size	Since the SEV could be improved by attaching more sensors later on, small sized gyros have the advantage

Table 6.1 - General Requirements for the Gyro

After an extensive research that took so many hours, finally a decision was made in regard of choosing a gyroscope to be used as a data collective device to sense the angular rate of the SEV while it is flying. Both the IDG-500 and the ISZ-500 by InvenSense were chosen to be installed. They implement each other to enhance the functionality of the system. When they are combined, they give a triple axis sensing. They were chosen because they really serve the needs of the SEV very well. The IDG-500 is a dual axis gyroscope, longitudinal and lateral, which will help in stabilizing the SEV in the sky. The ISZ-500 is a single axis gyroscope that implements the IDG-500; they have to be mounted in-plane for accurate triple axis sensing. Table 6.2 shows the specifications of both devices according to their user manuals by InvenSense. The power consumption suits the project; the analog outputs give the user the opportunity to decide how sensitive the gyroscope is needed to be. The voltage reference is connected to the Auto Zero function when using the 4.5 outputs in order to maximize the dynamic range of sensing to give the user a wider usable signal range.

To assist in the decision of choosing a gyroscope, we created a table to show the direct comparison between the ISZ-500 and the IDG-500. The specifications for these gyroscopes are shown below:

	ISZ-500	IDG-500
Number of Axis	Z	X/Y
Voltage Supply	3V	
Voltage Reference	1.35V	
Quiescent Supply Current	4.5mA	7mA
Power	13.5mW	21mW
Output per axis	Z _{OUT} , Z _{4.5OUT}	X _{OUT} , X _{4.5OUT} , Y _{OUT} , Y _{4.5OUT}
Dimensions	4 x 5 x 1.2 mm	

Table 6.2 - ISZ-500 and IDG-500 Specifications

6.2.2 Integration

The IDG-500 and ISZ-500 gyroscope are going to be integrated on the circuit board to send their readings to the microcontroller which is going to send the readings to the other microcontroller order to control the servos. The two outputs of the IDG-500 and the single output of the ISZ-500 send their readings as analog signals to the microcontroller. The IDG-500 is able to sense the rotations about both axes, X and Y, which allows the microcontroller to speak to the servos and tell them what must be done to stabilize the SEV, and the ISZ-500 is able to sense the rotations about the Z axis which will make the microcontroller send its signal to control the rudder. Figure 6.2 shows the integration of the IDG-500 and the ISZ-500 on the circuit board.

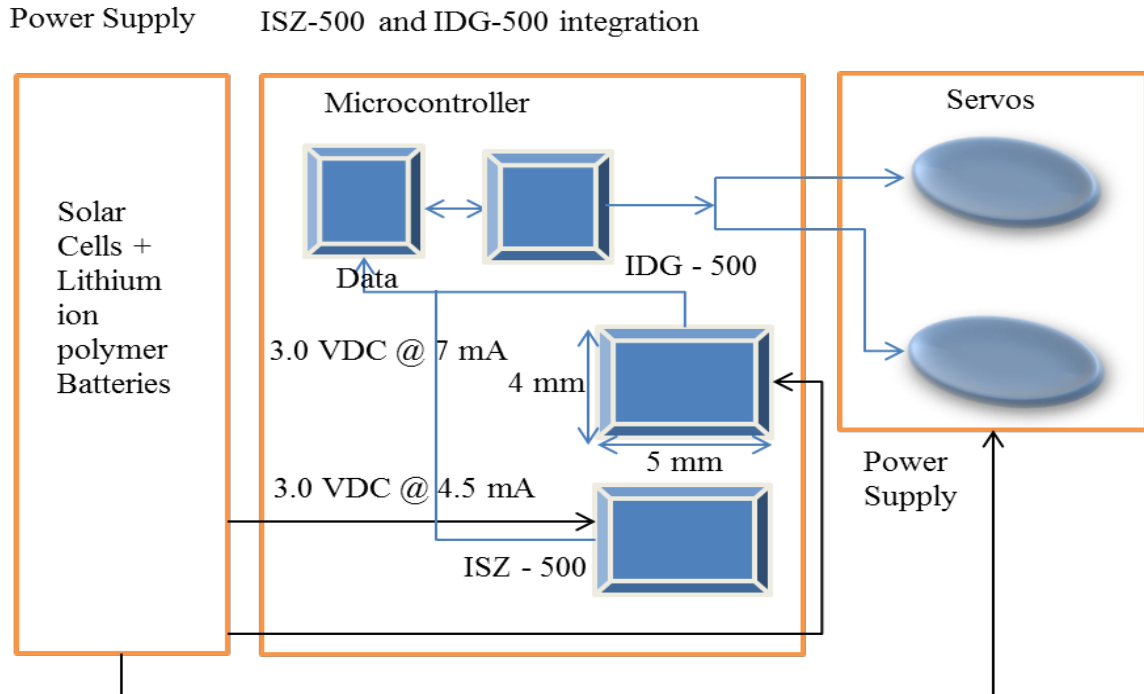


Figure 6.2 - the ISZ-500 and IDG - 500 Integration

6.3 Accelerometer

Combining an accelerometer with a gyroscope is reasoned to help stabilize the SEV. Since the gyroscope is measuring the rotational rates with respect to the longitudinal axis, X, and the lateral axis, Y, to deliver a message that tells how the SEV is behaving, the accelerometer is going to measure its movement with respect to the gravity which indicates that the accelerometer is very sensitive. Therefore, little vibrations are able to affect its readings. Hence, here comes the gyroscope helpful which is also affected by the vibrations but not as much as the accelerometer. A triple axis accelerometer measures the inertial forces with respect to the three axes, and most importantly it measures the acceleration of an object forward, backward, upward, downward, or at an angle. These measurements are transmitted to the microcontroller as digital signal or analog signal.

A very well programmed microcontroller uses both data sent from the gyroscope and accelerometer to analyze the behavior of the SEV. Then, the microcontroller transmits them to the other microcontroller which tells the servos how to adjust the SEV by controlling both the rudder and elevator. Figure 6.3 represents theoretically how an accelerometer works by having a ball inside a box; Figure 6.3a shows the forward movement; Figure 6.3b shows the backward movement; Figure 6.3c shows sudden drop; Figure 6.3d shows when an inclination occurs. Hence, the presence of the gravity force is there all the time because the position is given with respect to it.

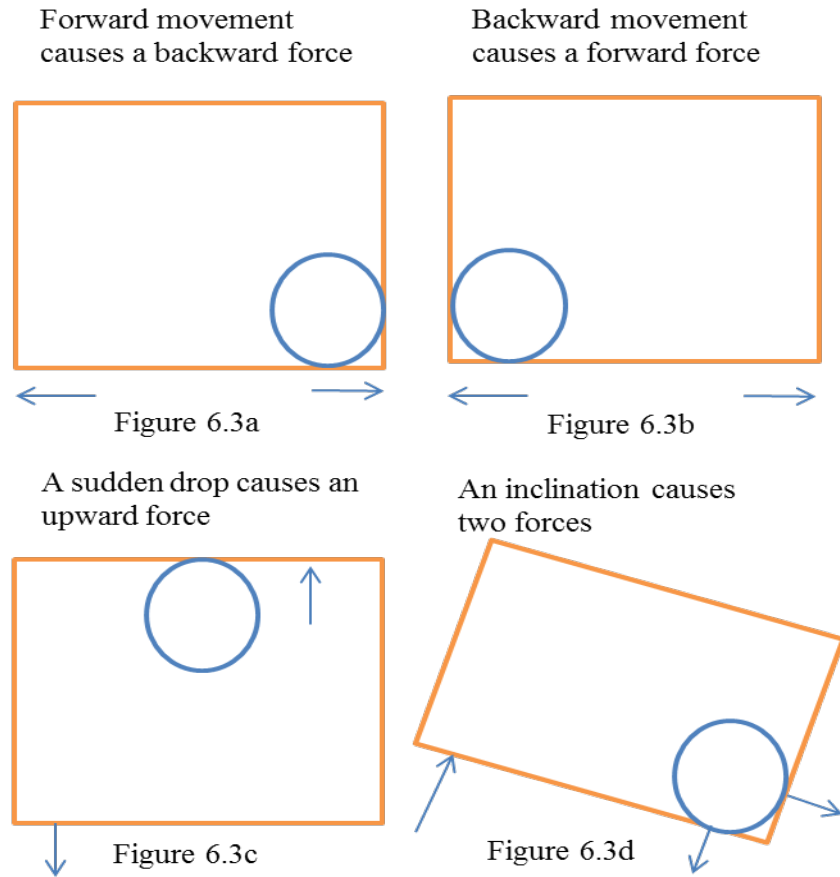


Figure 6.3 - Accelerometer Functionality

6.3.1 Basic Requirements

The requirements vary but focusing on the major needs such as the number of axes, power supply, and of course the dimensions of the accelerometer, determine which accelerometer is to be selected. Another factor of choosing the accelerometer is not to ignore the output, analog or digital signal; however, it is more related to the microcontroller. Since the microcontroller that is chosen supports both digital inputs and analog inputs, the searching task becomes easier. A triple axis accelerometer that consumes low power and very small in size is what the SEV requires. Thus, the MMA7361L, the low profile capacitive micro-machined accelerometer made by Freescale Semiconductor is chosen for this project. According to its data sheet by Freescale Semiconductor, Table 6.3 shows the features that the MMA7361L accelerometer has which qualify the basic requirements needed for the SEV.

MMA7361L	Basic Requirements
Number of Axes	3 axes, (X,Y,Z)
Power	Typical: 1.2 mW
Voltage	Minimum 2.2 V–Typical 3.0 V–Maximum 3.6

	V
Current	Typical 400 μ A – Maximum 600 μ A
Dimensions	3 mm x 5 mm x 1 mm
Sensitivity	800 mV/g @ 1.5g

Table 6.3 - MMA7361L basic requirements

6.3.2 Integration

The MMA7361L is going to be integrated on the circuit board beside the ISZ-500 and IDG-500 to send their readings to the microcontroller which is programmed to send the readings to the other microcontroller which analyzes the coming signal in order to control the servos. All of the devices together send their readings as analog signals to the microcontroller which is going to convert them into digital signals and analyze them. The MMA7361L is able to sense the movements in the three axes, X, Y, and Z with respect to the gravity. The data read and sent by the ISZ-500, IDG 500, and the accelerometer to the microcontroller allows the microcontroller to adjust the servos. The microcontroller gets the information from the devices to exactly figure out how the SEV is performing. Then it tells the servos what must be done to stabilize its performance.

Figure 6.4 shows the integration of the MMA7361L, ISZ-500, and IDG 500 on the circuit board as well as their measurements and the voltage supply needed for each device. The power supply for the microcontroller and the servos are left out because this figure is constructed to focus on the integration of the accelerometer and the gyroscopes.

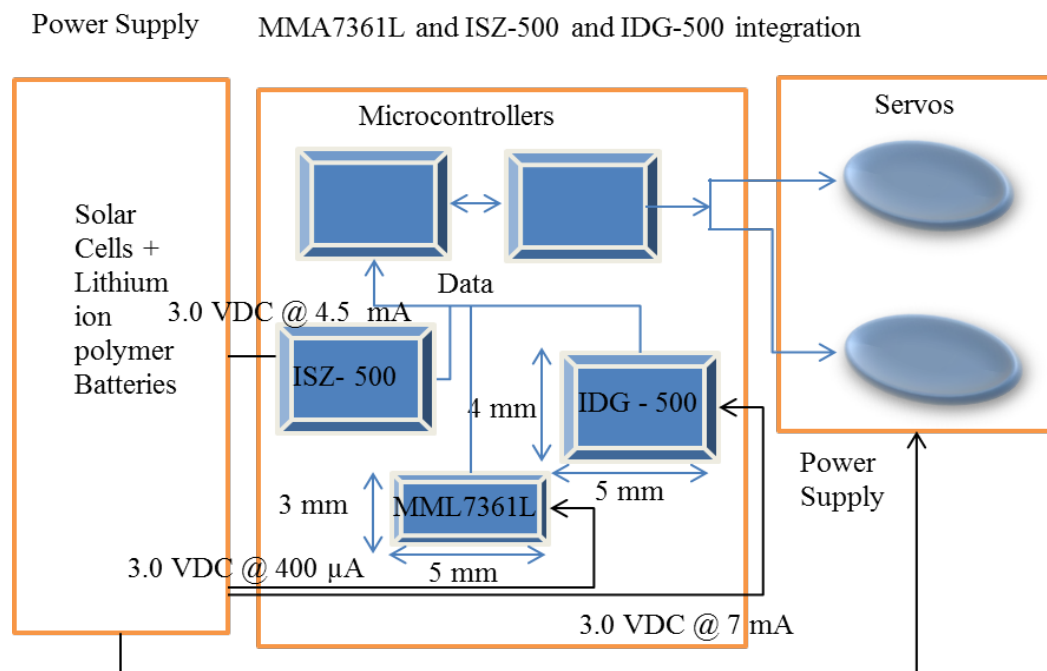


Figure 6.4 - MMA7361L and ISZ-500 IDG-500 integration

6.4 Barometric Pressure Sensor

The use of the atmospheric pressure sensor is pretty obvious for this project; it will be used to sense the barometric pressure at a specific location with respect to the pressure at the sea level which is approximately 101kPa or 29.92 inches Hg. The change in the pressure indicates either the object is above the sea level or below it. If the sensor readings are less than 101kPa, of course the SEV is flying above the sea level and vice versa.

According to the International Standard Atmosphere (ISA), the lapse rate is defined as the rate of temperature increase in the atmosphere with increasing altitude. For ISA the lapse rate near the ground is assumed to be $-6.5^{\circ}\text{C}/1000\text{m}$; it is also assumed that a constant lapse rate between 0 and 11Km altitude. Which indicates that the temperature decreases by 6.5°C for every 1000 m increase in altitude as long as the altitude does not exceed the 11 Km. According to Portland State Aerospace Society, table 6.4 describes all the constants which will appear in equation 6.1 regarding their symbols, values, and units.

Symbol	Value	Unit	Description
Po	101325	Pa	Pressure at zero altitude
To	288.15	K	Temperature at zero altitude
g	9.80665	m/s^2	Acceleration due to gravity
L	-6.5×10^{-3}	K/m	Lapse rate
R	287.053	J/(kg K)	Gas constant for air
Z	---	m	The altitude

Table 6.4 - symbol descriptions according to Portland State Aerospace Society

The derivation of the formula which relates altitude to pressure is derived by Portland State Aerospace Society as well. The result of the derivation appears in equation 6.1; it calculates the altitude due to the change in the atmospheric pressure which is symbolized as P.

$$Z = \frac{T_o}{L} \left(\left(\frac{P}{P_o} \right)^{-\frac{LR}{g}} - 1 \right)$$

Equation 6.1 – Altitude to Atmospheric Pressure

6.4.1 Basic Requirement

The basic requirements of the pressure sensor, BMP085 by BOSCH, come as shown in table 6.5. The table includes all the parameters needed for the SEV such as the voltage consumption, the average current consumption, the pressure range, and many more like its high precision. Its precision allows for more accurate altitude measurements to be reported to the ground station.

BMP085	Basic Requirements
Voltage Supply	1.8 – 3.6 V
Current Supply	Typical 12 μ A
Pressure Range	300 – 1100 hPa
Absolute Accuracy 700 – 1100 hPa Temp = 0 - +65°C, V= 3.3 V	Pressure: \pm 2.5 hPa, max Temp: \pm 2.5°C, max
Dimensions	5 mm x 5mm x 1.2 mm

Table 6.5 - Basic Requirements

6.4.2 Integration

Since the barometric pressure sensor is a digitized device, it is using the I²C serial port to transmit its data to the microcontroller. It is clear that in Figure 6.5 the sensors are divided into two sections, analog and digital. The analog sensors were discussed earlier which are the gyroscopes and the accelerometer; however, the barometric pressure sensor sends its data digitally. The following figure covers the navigation unit except for the GPS. It is placed on the right half side of the printed circuit board as discussed and shown in chapter 4. They are operated as shown in the figure at about 3-3.5 V which gives an advantage to make on voltage divider circuit to connect them in parallel with each respect to their voltage source.

Power Supply BMP085 and MMA7361L and ISZ-500 IDG-500 integration

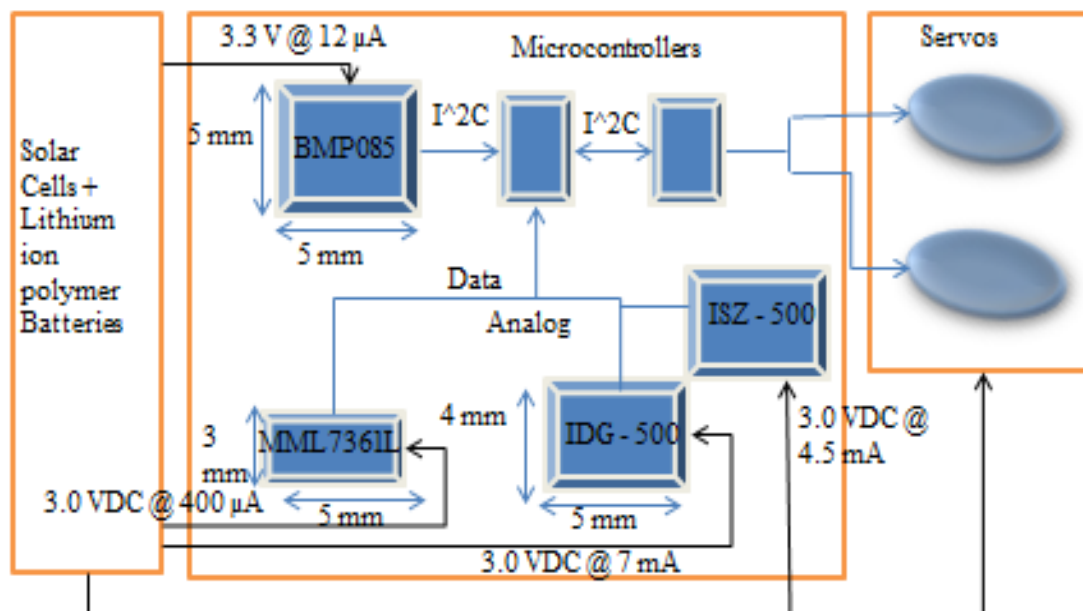


Figure 6.5 - the integration of all the sensors

Chapter 7: Communication

7.1 Machine to Machine Communication

Machine to machine communication system enables the machines to speak to each other by transmitting data from one end and receiving data on the other end. Such communication systems are used to control and track a specific system according to the information that been transmitted. Hence, the machine to machine communication system is not restricted to two machines only; the system can have multiple machines transmitting and receiving data. The applications of the system can vary, but the concept is built around the flow of data between two or more machines. Regardless of the type of the information being transmitted, it has to be going through basic steps. The first step is getting the data from a machine. The second step is sending the data through the network. The third step is reviewing the data on the other machine. The final step is to react according to the data. Thus, irrespective of the applications, machine to machine communication conceptually the same for all projects and designs.

Figure 7.1 of the machine to machine communication system demonstrates the concept by having a slave machine and a master machine. The figure shows that the slave machine collects the data and the master machine reacts to that information which is being transmitted through the network. In summary, the process consists of four steps: data collection, data transmission, revision of the data in order to response successfully.

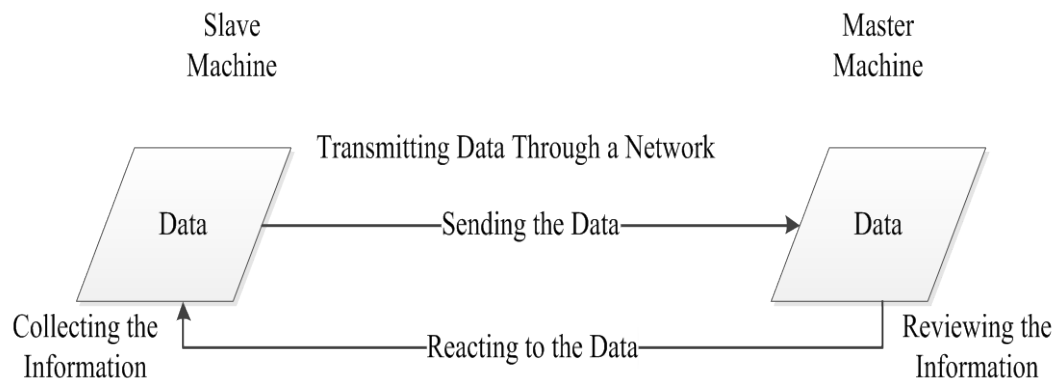


Figure 7.1 - Machine to Machine Communications

7.1.1 Data Collection

The Solar Extended Vehicle (SEV) contains many data collection devices such as an accelerometer, a barometric pressure sensor, a global positioning system (GPS), and two gyros. Each device collects data independently according to its functionality. The collected data for this specific project has to be transmitted to the microcontrollers to make use of them. The sensors and the GPS are slave machines; they only collect data to transfer them to the microcontroller.

7.1.2 Data Transmission

Once the data collected already, it is ready to be transmitted to the master machine which then is ready to be transmitted wirelessly to the ground controlling station by using a radio frequency transceiver to send the data and be able to receive some data back. Therefore, Zigbee is in charge for transmitting the data from the master machine on the SEV to the ground station.

7.1.3 Revision

The ground controlling station which is being presented as the master machine is connected wirelessly with the SEV; it receives data to revise them and determine whether to interrupt the system or not. If the data received indicates that the SEV is flying within the limit, it keeps the system running as programmed. On the other hand, if the data indicates the SEV is in danger the user should respond wisely by overriding the system to keep the SEV safe.

7.1.4 Response

Once the data is revised by the ground controlling unit, an appropriate response should be transmitted back whenever is needed. Since the SEV is programmed to survey a pre-determined region there should not be any worries. However, for safety purposes the response to be transmitted back to get the SEV to land as well as to complete the machine to machine communication system.

7.2 Global Positioning System (GPS)

The GPS is one of the data collective devices on the SEV that is used for tracking purposes. It is mainly used to keep track of the position and velocity with respect to time of the SEV. The recorded data is then transmitted to the microcontroller. The GPS is known to have some problems regarding its accuracy in the short run; however, it functions much better for a long period of time. The GPS market is crammed with many different systems that vary depending on the users need. For instance, some systems are designed to be installed in automobiles; others are designed to be installed in wrist watches or cellular phones for hikers. Obviously, the functionality of the GPS varies as well as the weight and the dimension of the system. In addition, to the power required to run the system and its sensitivity in detecting the signal.

As a result, Table 7.1 describes the basic requirements that are suitable for the SEV to be equipped with the GPS. Table 7.1 includes specific parameters regarding power, size, weight, sensitivity, and accuracy.

GPS	General Requirements
Power P=IV	Since the SEV is both solar and battery powered, the required GPS has to consume low voltage and low current.
Size	Since the SEV is limited to the size of the fuselage, the required GPS has to be very small in size.
Weight	Since the SEV is a sailplane, the required GPS is expected to be light to have a better power to weight ratio.
Sensitivity	Since the SEV is designed for surveillance purposes, the required sensitivity to detect the signals is expected to be very high.
Accuracy	Since the SEV is tracked, the required accuracy regarding velocity, altitude, and position has to be accurate enough for tracking purposes.

Table 7.1 - GPS General Requirements

7.2.1 GPS Requirements / Features

The general requirements were set previously, Table 7.1, yet different GPS's have arisen when the research was done. Therefore, the requirements had to be very specific to narrow down the outcome of the search. For instance, having said that the power has to be very low result in focusing on GPS's that consumes up to 3.6 volts and up to 30 milliamps at the standard mode which results in consuming up to 108 milliwatts. Size and weight are very important factors for the overall design as well as for choosing the appropriate GPS; thus, their importance comes right after the power. For the sake of having a compact GPS which consumes low power, the width and the length are decided to be no more than 30 millimeters each. Sensitivity and Accuracy are as important as the power usage to accurately track the SEV. Since the SEV is limited to a low power GPS's, the sensitivity and the accuracy are to be kept to what is offered by the market. However, the more accurate and the more sensitive GPS are to be compared next.

7.2.2 GPS Comparison and Selection

Comparing products is a crucial step in the research process which enhances the design with the best product in the market that fits needs of the project. Thus, Lassen iQ GPS and Copernicus™ GPS, two products that fulfilled the SEV requirements, are to be viewed to conclude with the best GPS for the SEV. Interestingly enough, both GPS's are produced by Trimble which is a company provide advanced location based solutions and well known for GPS technology. In the next paragraph the two choices are discussed and compared in order to find the one that satisfies the SEV design. According to their user manuals, these devices are very similar to each other and really hard to choose one on the other. Both of the devices meet the requirements on Table 7.1 as well as the requirements/features on Section 7.1.1. They both have 12 channels, and they both support three different protocols such as Trimble Standard Interface Protocol (TSIP), Trimble ASCII Interface Protocol (TAIP), and National Marine Electronics Association (NMEA). The software design decides which protocol needed for the SEV. The operational temperature is also the same, and they both have two serial ports to transmit

and receive. In addition, they also output position, velocity and time. The differences, as shown in Table 7.2, on the other hand, arise when their power, size, and weight are compared. Regarding the power, the Copernicus™ GPS receiver requires 0.05 milliwatts more than the Lassen iQ GPS receiver that indicates the Copernicus™ GPS requires either more voltage or more current to operate. Inversely, the Lassen iQ GPS receiver is heavier and bigger.

	Copernicus™ GPS	Lassen iQ GPS
Dimensions	19 mm x 19mm x 2.54mm	26mm x 26mm x 6mm
Power	< 94 mW	< 89 mW
Voltage	2.7 VDC – 3.3 VDC 3.0 VDC@ 31.3 mA → 93.9 mW	3.0 VDC – 3.6 VDC 3.0 VDC@ 27mA → 81 mW
Current	30.7 mA	27 mA
Antenna	Passive or active	Active only
Weight	0.06 Oz	0.2 Oz

Table 7.2 - Copernicus™ GPS vs. Lassen iQ GPS

As a result from the comparison, it is obvious how close they are regarding their differences. Power plays a major role in comparison between the devices, but it is not the only parameter in the previous table. Even if the power consumption is in favor of the Lassen iQ GPS receiver, the best choice that fits the SEV is the Copernicus™ GPS. It is better in weight and space; two parameters are very crucial to the overall outcome of the design.

7.2.3 Interface Design

The circuit board is going to be represented in a very simple manner to illustrate how they system is going to be implemented. Figure 7.2 shows only the GPS and the microcontroller. The chosen GPS is going to be soldered on the circuit board as well as the other components such as the microcontroller as it appears in the figure. Of course, the power is to be connected to all the components; however, it is not shown here that the microcontroller is powered since the main focus here is the GPS.

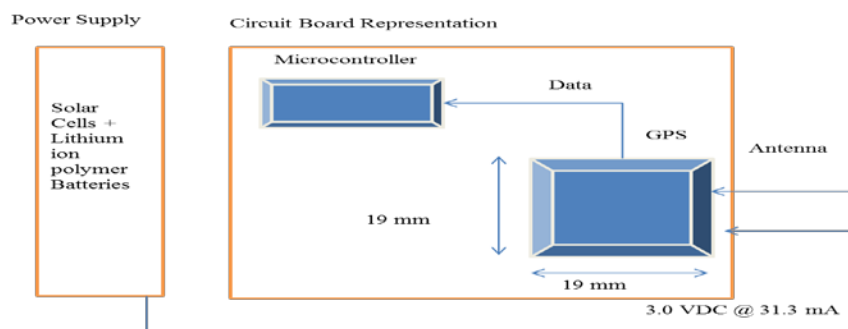


Figure 7.2 - GPS integration

The Copernicus™ GPS receiver consists of 28 pins; 8 pins are reserved; they are mostly not connected. The rest of the pins are used for other important connections such as pin number 12 the supply voltage, V_{CC} , which is between 2.7 – 3.3 V_{DC} . Pin number 0 is the ground, GND. Pins number 7 and 8 are called open and short respectively to determine that the external antenna detection circuit is either open or short. Pin number 11 is the reset which is active low. Pin number 16 is the standby mode that tells the GPS when to run. Pin number 19 is the timing signal at 1 Hz. The Copernicus GPS receiver as mentioned earlier supports three different protocols; Pin number 21, serial port A receive, and pin number 23, serial port A transmit, on the GPS receiver, are used to receive and transmit data as universal asynchronous receive and transmit (UART) pins. Pin number 5 LNA_XEN is a logic level output that is used to control power to an external circuit. The external circuit mainly used to connect the external antenna. Pins 7 and 8, the open/short pins, are there to be connected to an active antenna. Once the active antenna is connected, the open/short pins indicate that it is implemented with a short circuit protection.

Table 7.3 represents the condition of logic signals, tells when the antenna is open, short, normal or undefined.

	Short	Open
Antenna Open	1	1
Antenna Short	0	0
Antenna Normal	1	0
Undefined	0	1

Table 7.3 - Condition of Logic Signals

7.3 Zigbee

Going back to the data transmission, it is stated earlier that Zigbee is in charge of transmitting data collected by the various sensors and the GPS which are already sent to the microcontroller. Here comes the Zigbee to transmit all the data from the microcontrollers to the ground station using radio frequency waves. All the collected data through all the collective devices that are installed in the SEV such as the GPS, gyroscope, accelerometer, and barometric pressure sensor are sending their data to the microcontroller. The microcontroller sends them to the other microcontroller which is linked with the RF module to send the data to the ground station as the microcontroller programmed. Zigbee detects the ground station which is going to be connected to another Zigbee transceiver to communicate with the SEV effectively. The collective devices work independently, some data needs to be instantly sent to the ground station. For example, the data collected by the GPS has to be sent instantly for not losing track of the SEV and not letting the SEV to go out of range. However, other sensors do not have to report to the ground station as fast as the GPS unless a change occurs in altitude or the SEV itself such as rolling or turning. Figure 7.3 shows how the collective devices are connected with respect to the two microcontrollers.

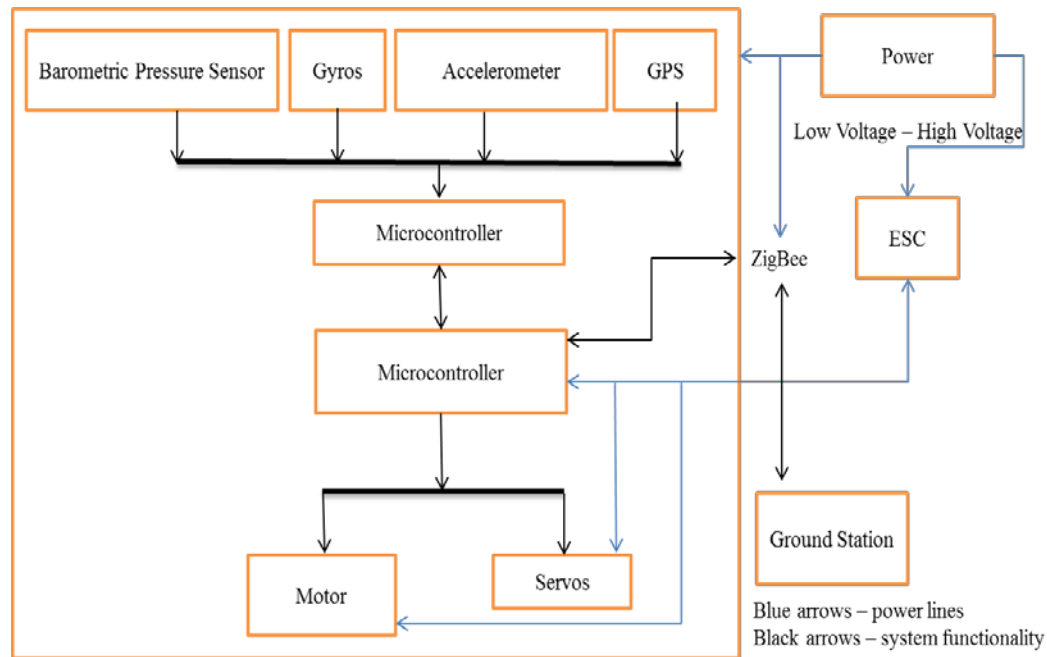


Figure 7.3 - the connections with the microcontroller

7.3.1 Basic Requirements

A system as the previously explained is obviously not sophisticated; however, it is what the SEV requires to operate. Of course, this basic idea of having collective devices being powered by solar cells and lithium ion polymer batteries is considerably not common which gives this project a unique touch. In addition, to make the SEV to be self-controlled system is another beautiful touch that the project is targeting. Zigbee at 900 MHz frequency band is able to transmit data at a rate of 250 Kbps. In addition, its outdoor range goes up to 6 miles, which serves the SEV very well. And most importantly its voltage supply is as low as 3.4 V. Table 7.4 shows the basic requirements of the RF module in more details.

Zigbee Pro 900	Basic Requirements
Outdoor Range	Up 6 miles
Data Rate	156.25Kbps
Receiver Sensitivity	-100 dBm
Supply Voltage	3 – 3.6 V
Transmit Current	180 mA @ 3.3 V
Receive Current	80 mA @ 3.3 V
Operation Frequency	900 MHz
Antenna Options	¼ wave wire antenna, U.FL connector, RPSMA connector
Dimensions	24.43 mm x 33.32 mm

Table 7.4 - Basic Requirements of the Zigbee

Chapter 8: Power Systems

8.1 General Overview

Our RQ-11 Raven autonomous UAV plane runs on an 11.1-volt rechargeable battery source that utilizes a brushless electric motor. Along with the battery system, it will be supplemented with 7.2 watts of solar power. The solar panels will not fully charge the battery system but it will allow for longer flight duration while the UAV is airborne. One of the major goals of the project was to have an airplane that could utilize the sun as an additional source of energy. In choosing our power scheme, we had several issues that we had to account for. The first of which we had to decide what our operating voltage was going to be. Secondly, we had to select the appropriate operating current of the solar module to charge our load. Generally, for charging any battery system it is to try to keep the total charging current to not any more than 10% of the total battery capacity.

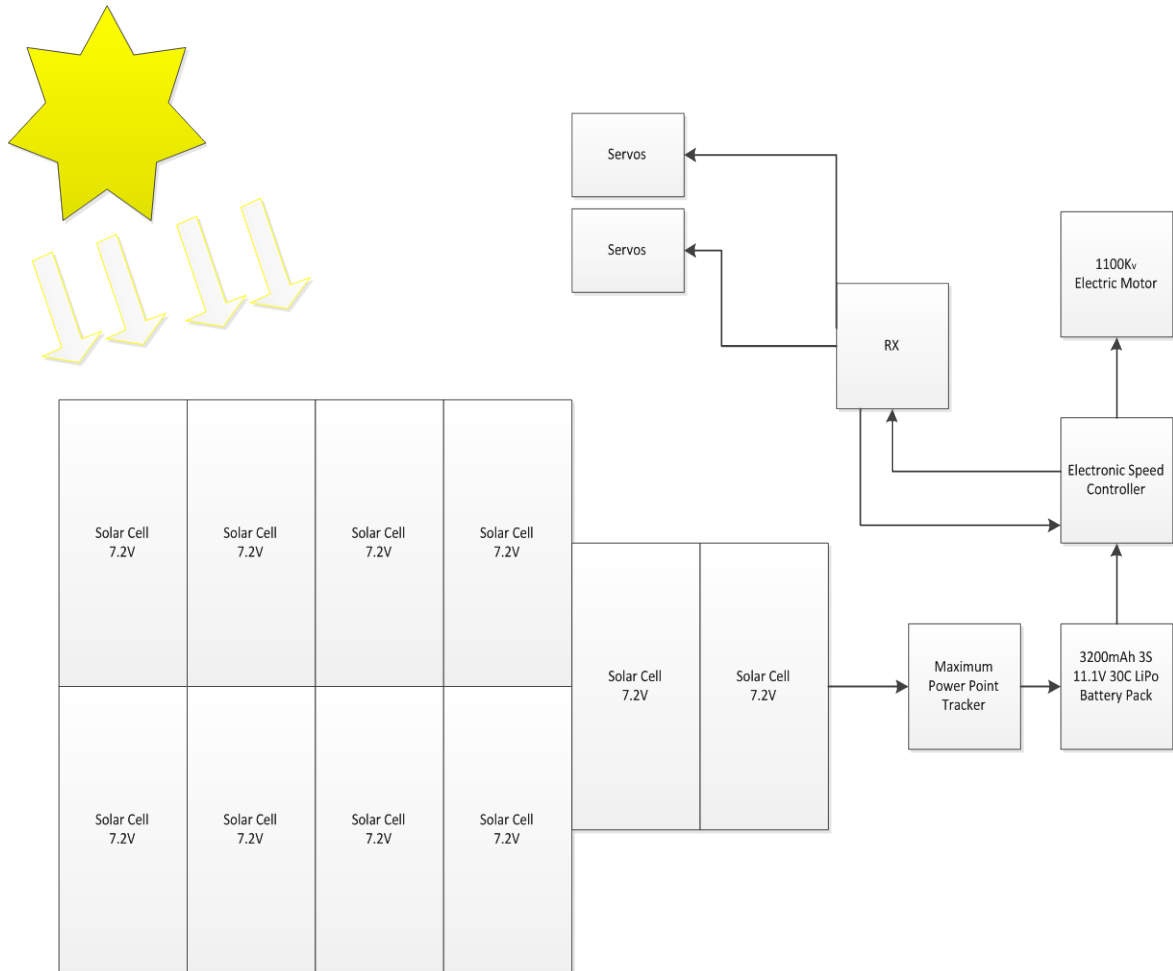


Figure 8.1 – Charging Circuit Overview

8.2 Solar Cells

All solar panels are made of silicon. Silicon is very abundant material on Earth. Just about every element or material that is found on Earth has a trace of silicon. There are three commercial types of solar electric panels readily accessible by consumers: single-crystalline, multi-crystal, and amorphous silicon cells. Single-crystal is also commonly known as Monocrystalline cells. Single-crystal cells are first purified and then are grown placing a seed crystal in the molten material. The seed acts as a catalyst for the remainder of the crystal growth. Single-crystal cells are the most efficient in converting light into useable electricity. In its current manufacturing technique, single-crystal cells are about 17% efficient in transferring solar energy to electricity. Single-crystal cells are currently the fastest in developing technology.

Multi-Crystal Silicon cells are also commonly known as polycrystalline. Multi-crystal silicon cells are about 12%-15% efficient in transferring solar energy to useable electricity. Multi-crystal silicon modules are graded on the size of the crystals and the number of impurities that are present after production. The higher the impurities, the less efficient the module. The larger the crystal wafers are more expensive to produce. Crystalline silicon cells are made from both mono- and poly-crystalline silicon cells. Crystalline silicon are the most expensive to produce because the very pure silicon that is required. The same material that is used in the production of computer processor is used in crystalline silicon cells. Amorphous silicon is also commonly known as thin-film PV. The advantages that thin-film PV modules have over their counterparts are lightweight, greater heat tolerance, better shading tolerance and the lower production cost. Amorphous silicon modules are about 8%-12% efficient in transferring solar energy to useable electricity. The biggest drawback of amorphous silicon modules are their lower power density as measured in watts generated per square unit. Amorphous cells lack efficiency but have flexibility, and crystalline cells have efficiency but lack flexibility.

8.2.1 Basic Requirements

The basic requirement that we had to meet in order for our project to be successful was that solar cells had to be high efficiency, light, and durable. With such a small surface area on the wing to attach the solar cells it was important to choose the right solar cell in terms of getting the most power from such a small area. The durability of the solar cells was also as important as the efficiency. We knew that having a plane meant that we would eventually have a few mishaps. Therefore, it was important to make sure that the solar cell could withstand the abuse during our test flights. As equally as important as the previous two reasons was that the solar cells had to be as lightweight as possible. When designing a radio-controlled airplane weight has to be taken into account. If the total weight of the plane is too much for it to create an equal or greater lifting force the plane will never leave the ground.

8.2.2 Selection

For our UAV project, we decided to use flexible thin cell solar panels that will be attached to the upper wing structure. The allowable area of the wing that we could use to mount our solar cell onto had approximately 525 area inches of space. During our initial project investigation, it was obvious that we had a lot different brands and styles that we could choose to use in our project. One such flexible solar panel was the SolMaxx Flex 7.2V 200mA. The SolMaxx Flex 7.2V 200mA has a 7.2V output that is rated at 100mA per cell. If the SolMaxx Flex 7.2V 200mA were to be used, we would need approximately four cells in a parallel/ series array. The total surface area that would be consumed by SolMaxx Flex 7.2V 200mA would be approximately 282.56 square inches and it would have an efficiency rated at 8.4%. The second solar cell that we took in consideration was the SolMaxx Flex 7.2V 100mA. The SolMaxx Flex 7.2V 100mA it also has a 7.2V output that is rated at 100mA per cell. In the case of the SolMaxx Flex 7.2V 100mA, we would have to use approximately eight cells with a surface area consumed of 330.72 square inches. Our third option is to use PowerFilm RC7.2-75. The PowerFilm RC7.2-75 would use approximately ten cells with a surface area consumed of 371 square inches. Table 8.2.2.1 shows the various specifications of each of the solar cells discussed above.

In our preliminary design considerations, we feel that the PowerFilm RC7.2-75 is our best option for several different reasons. In the initial research, that we did on the PowerFilm RC7.2-75 it was evident that this cell had the highest current output to weight ratio opposed to the SolMaxx Flex 7.2V 200mA. The SolMaxx Flex 7.2V 200mA may seem like a better option but in reality it is not. The other advantage that we have with using the PowerFilm RC7.2-75 is the total weight of the array. Using the PowerFilm RC7.2-75 our total array weighs only 2 ounces opposed to the SolMaxx Flex 7.2V 200mA weighing at 9.9 ounces with less power output. The other clear advantage with using the PowerFilm RC7.2-75 is the total thickness of only 0.2mm. This will prove to be very important when we are trying to keep the UAV as aerodynamic as possible.

Panel	SolMaxx Flex 7.2V 100mA	SolMaxx Flex 7.2V 200mA	PowerFilm RC7.2-75
Dimensions:	10.6" x 3.9"	10.6" x 6.9"	10.6" x 3.5"
Weight:	1.1 oz	1.9 oz	0.2 oz
Total Weight:	9.9 oz.	7.6 oz.	2 oz.
Thickness:			0.2mm
Voltage:	7.2V	7.2V	7.2V
Power Output:	0.72W	0.72W	0.72W
Total Output:	400mA 14.4V	400mA 14.4V	500mA 14.4V

Operating Current:	0.100 amps	0.100 amps	0.100 amps
Price:	\$20.95 ea.	\$37.75 ea.	\$27.45 ea.

Table 8.2.2.1 – Solar Panel comparison

Figure 8.2.2.1, 8.2.2.2 and Figure 8.2.2.3 shows each of the configurations and placement on the 51-inch wingspan of our UAV project. In Figure 8.2.2.1, it clearly shows that we can install ten cells on our wing. Thus giving us a total capacity of 7.2 watts with a resistance of about 17.1Ω. The disadvantage to this setup is the total weight of the SolMaxx Flex 7.2V 100mA comes in at 9.9 ounces.

Solar Cells SolMaxx Flex 7.2V 100mA	Solar Cells SolMaxx Flex 7.2V 100mA	Solar Cells SolMaxx Flex 7.2V 100mA	Solar Cells SolMaxx Flex 7.2V 100mA	Solar Cells SolMaxx Flex 7.2V 100mA
Solar Cells SolMaxx Flex 7.2V 100mA	Solar Cells SolMaxx Flex 7.2V 100mA	Solar Cells SolMaxx Flex 7.2V 100mA	Solar Cells SolMaxx Flex 7.2V 100mA	

Solar Cells

SolMaxx Flex 7.2V 100mA

Figure 8.2.2.1 – SolMaxx Flex 100mA

Figure 8.2.2.2 incorporates four SolMaxx Flex 7.2V 200mA solar cells. At first glance it seems like the best option but it really isn't since the total weigh comes in at 7.6 ounces.

Solar Cells SolMaxx Flex 7.2V 200mA	Solar Cells SolMaxx Flex 7.2V 200mA	Solar Cells SolMaxx Flex 7.2V 200mA	Solar Cells SolMaxx Flex 7.2V 200mA
--	--	--	--

Solar Cells

SolMaxx Flex 7.2V 200mA

Figure 8.2.2.2 Solmaxx Flex 200mA

On the other hand, the PowerFilm RC7.2-75 will give us the capability of installing ten cells as shown in Figure 8.2.2.3. This would give us total power output of about 7.2 watts and only weighing in at 2 ounces.

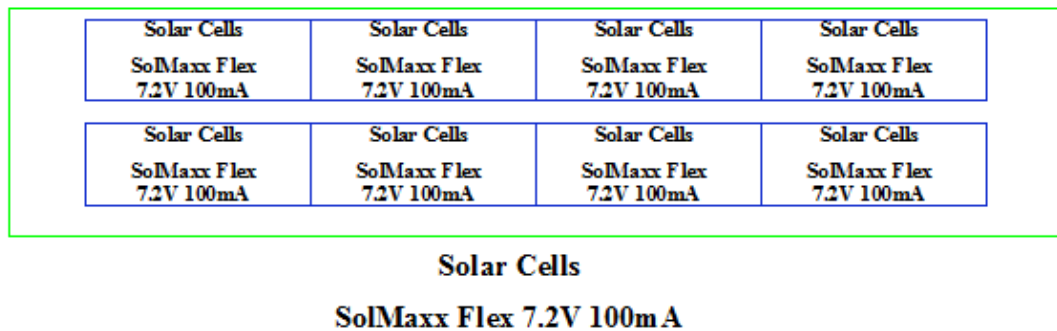


Figure 8.2.2.3 RC7.2

8.2.3 Interface Design

To maximize the total output of the PowerFilm RC7.2-75 solar panels we will have to arrange them in a series/parallel circuit. This is being done by having four groups of two cells. Figure 8.2.3.1 shows the exact circuit that we are implementing in our design of the solar array. As stated previously the total power output of the array will be approximately 7.2 watts with a total resistance of 28.8Ω . It is imperative when connecting the solar modules that we recognize the correct polarity of when we interconnect them. If we were to connect the panels to the battery with an incorrect polarity, we are risk of damaging the cells or it may cause the battery to explode causing bodily harm or even death. With each of the cells being rated at 9.9 volts each we will need to have them in series/ parallel connection to accommodate for the 19.8 volts needed to charge our battery system. The minimum wire size that we will use for our project will be 24 gauge copper wire. At the end of the circuit, there will also be a reverse blocking diode such as a 1N5817 to prevent any back feed from the batteries to the solar panels. By using a blocking diode at the end of the circuit, it will also prevent any ground fault in the system thus potentially saving the solar cells from any residual feedback from the system.

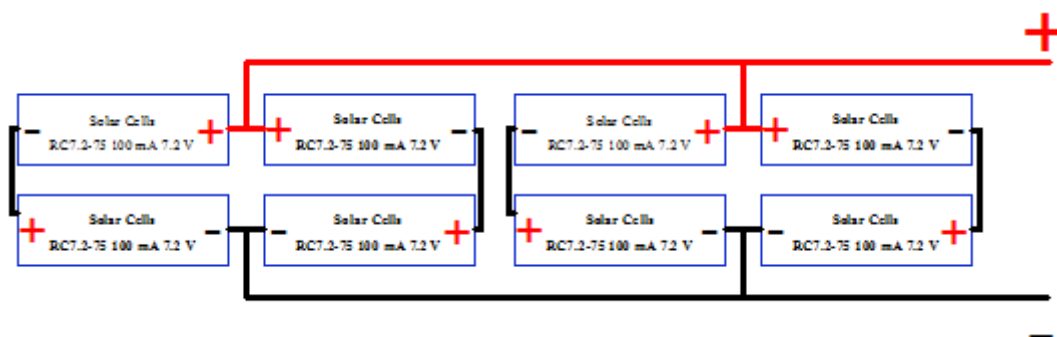


Figure 8.2.3.1 - Solar Array Configuration

8.3 Battery

The selection of the correct motor and the electronic speed controller to be used in our UAV project was paramount in our power system design process. We had to choose whether the sailplane would be driven under internal combustion type of motor or an electric driven motor. UAVs power systems can vary significantly based on its application. Solar powered UAVs are currently been researched in order to provide "perpetual" motion with the aid of hi-density polymer batteries for night motion.

8.3.1 Basic Requirements

The battery pack that we have chosen for our UAV had to match with the correct voltage input of the electric motor that we selected. During our preliminary design criteria, we selected an E-flite EFLB1040. The E-Flite EFLM4010A has 2.10 amp current draw at 10 volts when the motor is at an idle speed. The max current draw that we will ever see from the E-Flite EFLM4010A is 42 amps. Therefore, it was imperative that we chose the correct battery pack size and the correct electronic speed controller. The battery pack that we choose for our project is an E-flite EFLB1040. The E-flite EFLB1040 is a 3200mAh 3S 11.1V 30C Lithium-Polymer battery pack.

8.3.2 Selection

The recent demand for electric vehicles, and the need for storage of electricity from wind or solar generated power, have all caused a recent push for improved battery technology. For our senior design project, we had a few different choices to make on what type of battery that we going to use for our autonomous UAV. Through the research that we did, we found that there are several different types of batteries in the current market. The major different types of batteries used in radio-controlled airplanes are Nickel-Cadium (NiCad), Nickel-Metal Hydride (NiMH), Lithium-Polymer (LiPo), and LiPo hybrids. Nickel-Cadium (NiCad) has many advantages and limitations as a stable power source for our project. Some of the advantages of Nickel-Cadium are it has a High number of charge/discharge cycles, good low-temperature performance, and has a simple storage and not subject to regulatory control when transporting it. Some of the disadvantages of Nickel-Cadium battery packs are its high memory effect that in turn requires periodic full discharge cycles, it is environmentally unfriendly, and it has a relatively low specific energy compared to other battery types. Nickel-Metal Hydride (NiMH) has many advantages and limitations as a stable power source for our project. Nickel-Metal Hydride battery packs are 30%-40% higher capacity than the standard Nickel-Cadium (NiCad) battery packs. Nickel-Metal Hydride battery packs are also less prone to memory effects, it too has a simple storage and is not subject to regulatory control, it is environmentally friendly, and the contents are recyclable. Some of the limitations of Nickel-Metal Hydride battery packs are it has a short life cycle, it requires a complex

charging algorithm, it has a high self-discharge rate, and it is very susceptible to overcharging which in turn can damage the battery pack.

In particular, we were deciding between two different types of batteries, LiPo hybrids, and the Lithium-Polymer (LiPo). The differences between the battery packs are listed below in Sections 8.4.2.1 and 8.4.2.2. LiPo battery packs are great because they can store 350% more energy than a typical NiCad battery pack and weighing 10%-20% less. LiPo batteries also do not develop memory or voltage depression characteristics like NiCad batteries.

8.3.2.1 LiPo Battery Packs

A true LiPo battery does not use a liquid electrolyte but instead uses a dry electrolyte polymer that resembles a thin plastic film. This film is sandwiched between the anode and cathode of the battery allowing for ion exchange as seen in Figure 8.4.1, thus the name lithium polymer. This method allows for a very thin and wide range of shapes and sizes of cells. The problem with true LiPo cell construction is the ion exchange through the dry electrolyte polymer is slow and thus greatly reduces the discharge and charging rates. This problem can be somewhat overcome by heating up the battery to allow for a faster ion exchange through the polymer between anode and cathode, but is not practical for most applications. This is remedied by keeping the battery pack around 60 degrees Celsius to allow for better conductivity. Typical lithium polymer battery packs range in size between 20mAh up to 4000mAh and have an input voltage range of 3V-4.2V per cell. Lithium polymer battery packs have twice the energy density of a nickel-cadmium battery. A typical Lithium-Polymer battery pack may be stated as the following, 2200mAh 4S1P 20C 14.8V LiPo Battery. The battery capacity is denoted by the 2200 mAh rating, the number that precedes the "S" denotes the total number of cells in series, and the "P" denotes to the number of cells that are wired in parallel. The number that precedes "C" is the maximum discharge rate in amps that the battery pack can withstand at any given moment. In the above stated example, 2200mAh 4S1P 20C 14.8V LiPo Battery, the total capacity of the battery pack is 3200 milliamp per hour that consist of four cells wired in series to produce 14.8V at a maximum discharge rate of 20 amps.

8.3.2.2 LiPo Hybrid Battery Packs

Most of the batteries sold today for radio-controlled models are a hybrid lithium polymer battery. The correct name for this type of battery is lithium-ion polymer, but the battery world of today simply calls them lithium polymer even though they are not a true dry type LiPo battery. By introducing a gelled electrolyte into the polymer, the ion exchange rate is improved immensely. Since the electrolyte is gelled, there is less chance of leakage, but it is still flammable. LiPo hybrids are not as dangerous as Li-Ion's but they can still catch fire or explode if over charged, shorted, or punctured. When first introduced, LiPo batteries were more expensive than Li-Ion because they are more difficult to manufacture. Fortunately prices have dropped substantially since they have become as, if not more popular than Li-Ion battery technology. This holds especially true

for electric powered RC aircraft. LiPo hybrids use the same flat cell structure as their dry counter parts meaning they have the same flexibility with sizes and shapes allowing for very specialized shaped battery packs perfect for use in our autonomous UAV project.

8.3.3 Interface Design

Lithium-ion polymer battery packs are an advanced technology battery pack that requires a small deviation in charging voltages. LiPo battery packs need a constant charging voltage opposed to NiCad or NiMH. The voltage applied to a battery always has to be higher than the basic voltage of the battery under charge if you want it to charge correctly. Each cell in the battery pack will have a nominal voltage of 3.7V and 4.2V when fully charged. The basic algorithm is to charge at constant current 0.2 C to 0.7 C until the battery reaches $4.2 V_{pc}$ (volts per cell). Meanwhile a fully discharged 3-cell LiPo battery pack will have a reading of 12 volts across the entire battery pack The 11.1v represents the nominal voltage of three individual LiPo cells wired in series $3.7v \times 3$. The average cutoff voltage of the electronic speed controller is 3.0-3.2 volts. This allows for a buffer in the battery system so that the battery is not completely discharged thus avoiding damage to the battery. Also, you should never dead short a LiPo pack, even if only for an instant, as the large amount of energy stored in the small package can catch fire quite quickly and/ or cause bodily harm. LiPo batteries are nothing to play with like older NiCad or NiMH batteries.

The battery pack that we chose for our project is an E-flite High-Power Series Lithium Polymer EFLB1040 battery. This particular battery pack is commercially available and is very common with RC enthusiast. Most of all it is a perfect match for the E-Flite EFLM4010A electric motor that we chose for our project. The E-flite EFLB1040 is a 3200mAh 3S 11.1V 15C Lithium-Polymer battery pack. The E-Flite EFLM4010A has a maximum 50C discharge rate where the maximum current draw that we will ever see from the E-Flite EFLM4010A is 42 amps. Each of the individual Cells are rated at 3.7V with a total pack voltage of 11.1V. Thus, the total amperage capacity of the battery pack would remain at 3200 mAh with a 15C discharge rate. The configuration in Figure 8.3.3.1 would result a pack rated for 11.1V @ 3200 mAh with a maximum continuous discharge rate of 30 amps. The E-Flite EFLM4010A is 1.4 in x 1.60 in x 0.20 in and weighs only 4.3 ounces.

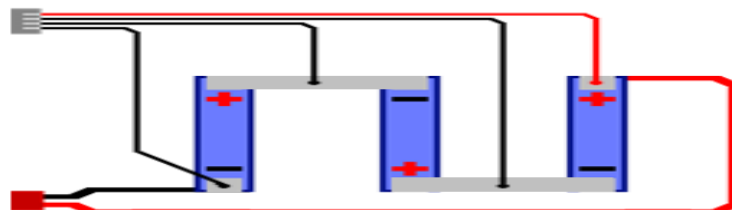


Figure 8.3.3.1 – EFLB32003S30 Lithium-Polymer Battery

8.4 Maximum Power Point Tracker (MPPT)

8.4.1 Charging Circuit

Solar panels give a great potential as an energy harvesting power source all they need is a battery system to store all of the energy that was collected. In order for the batteries to be charged there has to be a charge controller that regulates the solar panel voltage to match the ideal charging voltage of the battery system. The ideal device that controls the output voltage of the solar panels to the input voltage of the battery system is called a maximum power point tracker. Maximum power point trackers work by 'tracking' the voltage and current curve of a solar panel so that the total Power (Voltage * Current) is maximized. What this means is that as the light illumination level changes the voltage and the current must be carefully corrected to meet the correct output voltage of the circuit. An interesting correlation exists in solar panels output voltage. Whatever the existing light level illumination is the output voltage will always remain the same. Figure 8.5.1 uses the green lines to show the I-V curve of the panel for a given light condition. As the light increases, the voltages stays sort of the same but the amount of current you can draw goes up. If you can keep the DC/DC converters operating on the red line that is the maximum power. Figure 8.4.1.1 portrays the current-voltage curve and the power-voltage curve of the panel for a designated light condition. The maximum power is indicated by the red line.

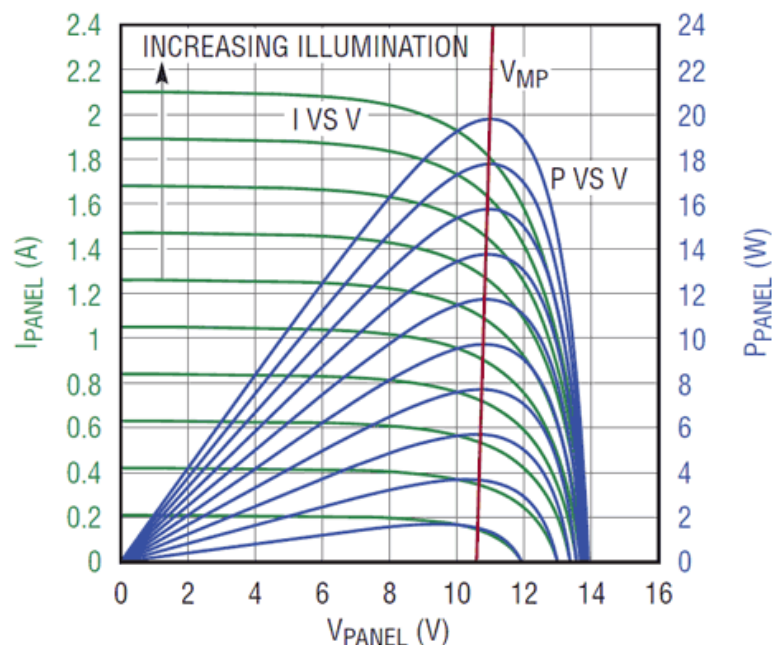


Figure 8.4 1.1 - Current/Voltage Curve and Power/Voltage Curve
Reprinted with permission of Linear Technology

8.4.2 Basic Requirements

The basic requirements that we were looking for in our charging circuit was that it had to have a protection circuitry to protect our lithium polymer battery pack from being overcharged. Lithium polymer batteries are very different from NiCad or NiMH battery packs. Lithium polymer chargers are specifically designed to charge lithium batteries thus certain aspects of a charger have to be taken into account. For instance, a lithium polymer charger has to be set with the correct number of cells and the correct float voltage and these voltages cannot exceed 3.0V-4.2V per cell +/- 0.05V per cell when the battery pack is fully charged. As shown on Figure 8.4.2.1, lithium polymer batteries have to have a specific charging algorithm in order for it be fully charged. Stage 1 of a lithium polymer battery is between 0.5C and 1C and typically takes about three hours. The main objective of the stage 1 charging is that it provides a high current and thus allowing the battery pack to reach its threshold voltage. The charge level at this point is about 70 % of the battery pack. Stage 2, saturation charge, of a lithium polymer battery is when the battery pack has reached its upper voltage threshold and then the current begins to drop and level off at about 3% of the nominal charge current. Stage 3 is rarely used but when used it provides the topping charge. Stage 3 is typically used once every 500 hours of use. Lithium polymer battery packs do not need a trickle charge or a float charge because it is unable to absorb an overcharge. If the battery pack was to trickle charge, it could cause plating of metallic lithium, a condition that renders the battery pack unstable. The second requirement that our charging circuit had to have was low voltage cutoff set point. Over discharging can also cause copper plating that leads to the internal shorting of the lithium polymer cells. The third requirement was that our charging circuit had to be compatible with our input source. In our case, the source was an array of solar panels. Specifically 7.2 watts of solar cells which in turn translated to about 19.8 volts of useable input. The fourth requirement that we were implementing into our circuit was a maximum power point tracker. In brief, a maximum power point charger is a controller that boosts the current and voltage of the charging state of the battery while using the most of the available power of the solar panel. All of these requirements would have to be met while maintaining the highest efficiency possible. Figure 8.4.2.1 shows the charging stages for a lithium polymer battery. Stage 1 of the battery is between 0.5C and 1C and typically takes about three hours. During stage 2 the battery pack has reached its upper voltage threshold and the current begins to drop and level off at about 3% of the nominal charge current. Stage 3 is rarely used.

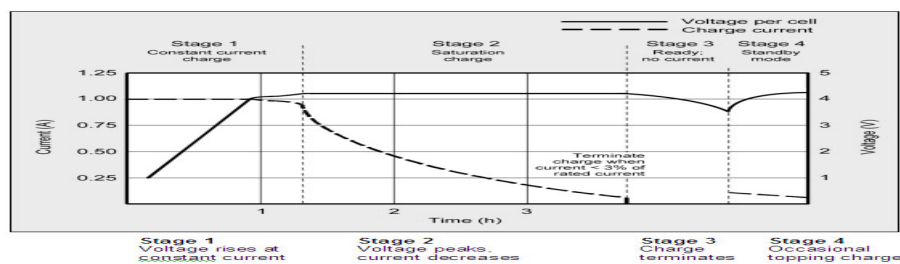


Figure 8.4.2.1 – The Charging Stages for A Lithium Polymer Battery
Reprinted with permission of Linear Technology

8.4.3 Design

Our maximum power point tracker will incorporate a Linear Technology LT3652 integrated circuit chip that was solely designed to be a power tracking battery charge for solar powered applications. The LT3652 is a complete monolithic, multi-chemistry battery charger that addresses high voltage applications in a compact design. The key features of the LT3652 are wide input voltage range of 4.95V to 32V with an absolute max voltage of 40V, programmable external charge current of up to 2 amps, resistor programmable float voltage up to 14.4V with 5% charge current accuracy and 0.5% float voltage reference accuracy. The LT3652 also has an input voltage regulation loop that reduces the current if the input voltage falls below a pre-programmed voltage level set by the voltage divider resistors. The LT3652 also contain provisions for battery temperature. This implemented by using a thermistor during the charging cycle. As seen in Figure 8.4.3.1 the LT3652 tracks maximum power point to 95% or greater when the charging voltage is set around 12.6V. At higher charging voltages it increase to 98% or greater.

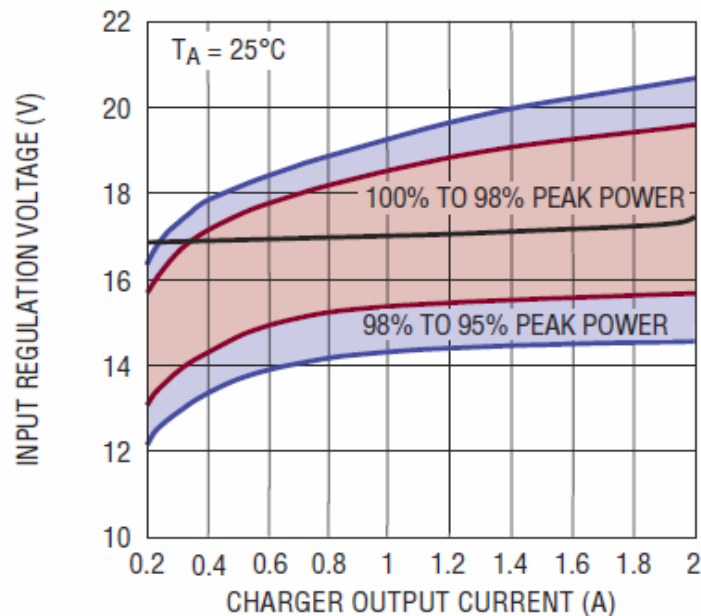


Figure 8.4.3.1 – LT3652 MPPT Current Tracking
 “Reprinted with permission of Linear Technology”

8.4.4 Design Integration

The LT3652 is offered in two separate packages from Linear Technology, the DD package and the MSE package. For our maximum power point tracking charging circuit, we will be using the MSE package. The LT3652 has twelve pins with the right combination of resistors, capacitors, and diodes we can have the LT3652 output the correct voltage and amperage that is needed for our charging circuit as shown in Figure 8.4.4.1.

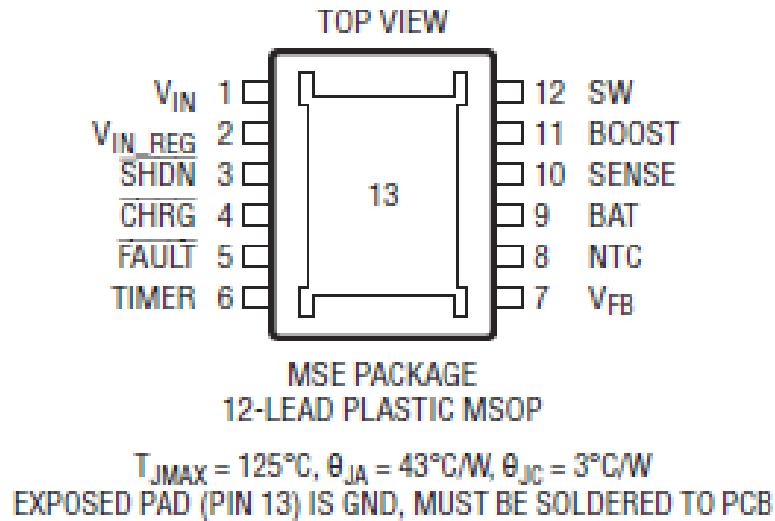


Figure 8.4.4.1 – LT3652 DD Package
“Reprinted with permission of Linear Technology”

Charge current programming is set by choosing an inductor sense resistor. For our particular circuit that we are designing the total expected max current that we would see from the circuit is 462.7mA. The expected value for R_{Sense} would be a resistor with an approximate value of 0.2161Ω as seen in Equation 8.4.4.1 and Figure 8.4.4.2. Using a resistor divider is needed to program the desired float voltage, $V_{\text{BAT(FLT)}}$, for the battery system. In particular, resistors R_{FB1} and R_{FB2} will have to have the correct values to set the 12.6-volt float charge needed in the lithium polymer battery pack.

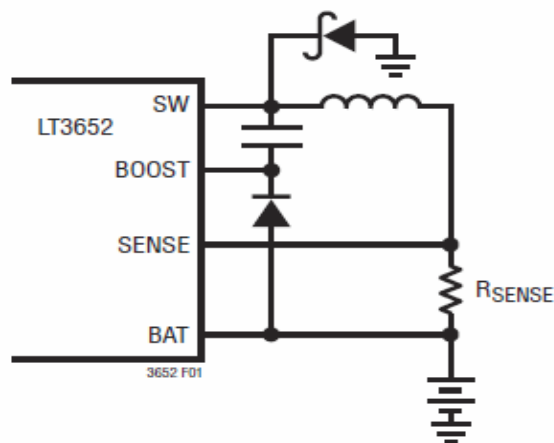


Figure 8.4.4.2 - Programming Maximum Charge Current Using R_{SENSE}
“Reprinted with permission of Linear Technology”

In order to achieve the 12.6V needed we had to use Equation 8.4.4.2 and Equation 8.4.4.3 to yield the correct resistors to be used in our design as shown in Figure 8.4.4.3.

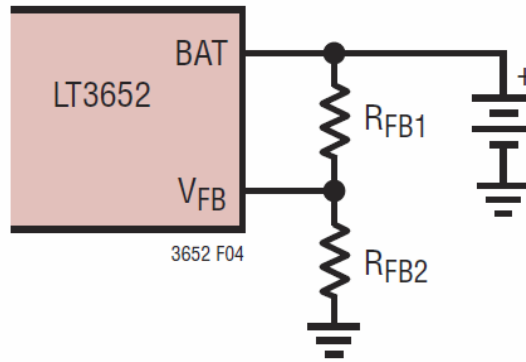


Figure 8.4.4.3 - Feedback Resistors from BAT to V_{FB} Programming Float Voltage

“Reprinted with permission of Linear Technology”

The LT3652, as shown in figure also contains a voltage monitor pin that enables it to monitor the minimum amount of voltage coming into the MPPT. The input supply voltage regulation is controlled via the voltage divider resistor R_{IN1} and R_{IN2} . An operating supply voltage can be programming by monitoring the supply through the resistor divider network. This is done by having a ratio of R_{IN1}/R_{IN2} for a desired minimum voltage. In order to achieve the 12.6V needed we had to use Equation 8.4.4.4 to yield the correct resistors to be used. The inductor has its own equation to yield the correct value as seen in Equation 8.5.3.5. ΔI_{MAX} is the normalized ripple current though the inductor. The LT3652, as shown in Figure 8.4.4.4, also contains a timer termination in which the battery charge cycle is terminated after a preset time. Timer is engaged when a capacitor is connected from the “TIMER” pin to ground otherwise it is directly connected to the ground. A typical timer termination is set to three hours but for our project design we have eliminated this feature from the final design therefore it will be connected directly to ground.

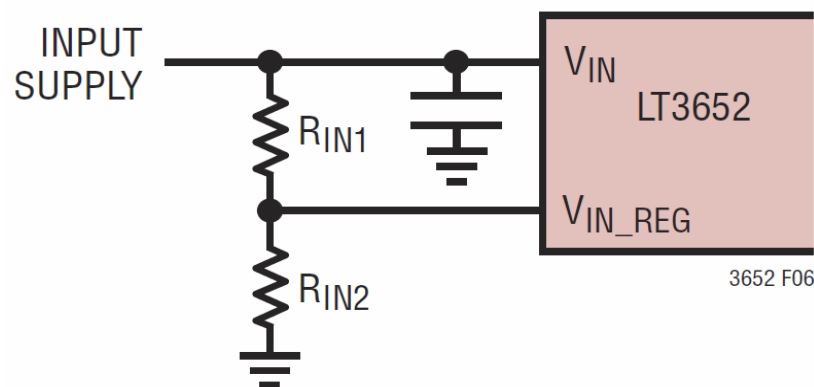


Figure 8.4.4.4 - Resistor Divider Sets Minimum V_{IN}

“Reprinted with permission of Linear Technology”

The table below was described earlier to explain the previous figures. They are introduced in Table 8.4.4.1 to represent them in an organized manner.

	Principle Equation	Result
Equation 8.4.4.1	$R_{SENSE} = 0.1/I_{CHG(MAX)}$	$R_{SENSE} = 0.2161 \Omega$
Equation 8.4.4.2	$R_{FB1} = (V_{BAT(FLT)} * 2.5 * 10^5)/3.3$	$R_{FB1} = 943 \Omega$
Equation 8.4.4.3	$R_{FB2} = (R1*(2.5*10^5))/(R1*(2.5*10^5))$	$R_{FB2} = 340 \Omega$
Equation 8.4.4.4	$R_{IN1}/R_{IN2} = (V_{IN(MIN)}/2.7) - 1$	$R_{IN1}/R_{IN2} = 12.185 \Omega$
Equation 8.4.4.5	$L = (10 R_{SENSE}/\Delta I_{MAX}) * V_{BAT(FLT)} * [1 - (V_{BAT(FLT)} / V_{IN(MIN)})]$	$L = 38 \mu H$

Table of Equations 8.4.4.1 – Resistors/Inductor Calculations

The LT3652, as shown in Figure 8.4.4.5, also has status and fault pins that can communicate back a charge or a fault status. The “CHRG” status pin states when the charger is delivering current larger than a C/10 rate or 1/10th of the predetermined charge current. The “FAULT” status pin states when the charger has a bad battery or a NTC fault. Table 8.4.4.2 shows the status pin states that the LT3652 will communicate back through the incorporated diodes.

STATUS PINS STATE		CHARGER STATUS
CHRG	FAULT	
OFF	OFF	Not Charging — Standby or Shutdown Mode
OFF	ON	Bad Battery Fault (Precondition Timeout / EOC Failure)
ON	OFF	Normal Charging at C/10 or Greater
ON	ON	NTC Fault (Pause)

Table 8.4.4.2 – Status Pins State
“Reprinted with permission of Linear Technology”

Below in Figure 8.4.4.5 is the charging circuit that was designed from the setup equations and diagrams. On the left side it shows the input voltage which is the output voltage of the solar panels. The voltage coming in is going through several diodes to prevent any feedbacks. It is also going through several voltage dividers to bring down the voltage to charge the battery which is located at the right bottom corner of the figure. The LT3652 takes the voltage from the solar panels and tracks its maximum current flow to charge the battery with the highest possible current flow. The purpose of the charging circuit is to charge the battery most efficiently through the LT3652.

rudder and throttle. It also supports “fly-by-wire”, which allows the user to tell the S.E.V. where to go and the ArduPilot will fly the plane; this is very important when inexperienced flyers take control.

We are going to need an Integrated Development Environment (IDE) to build and load this code to the microcontroller. The program we have chosen to do this is the Arduino IDE. This IDE is a simple and easy platform for any level of experience user to use and implement into any project. Arduino also offers an array of FPGA’s to test and develop code. We will be using a UAV board that is offered from Sparkfun, to help with testing and calibrating the code. In the following sections of this chapter we will discuss how the different parts of the code will work in conjunction with the microcontrollers and sensors. That is, how the pitch, roll, and altitude are obtained and utilized throughout the code.

9.2 General Breakdown of the ArduPilot Code

To use ArduPilot, the program needs to be configured because not all UAVs are created equal. These configurations will be done through the Config.h tab; after the program has been open using Arduino IDE. This file is for you to tell the program about your UAV. For example, what type of tail configuration you have, if you going to use ailerons or not, and any offset calibration will be done here. After these configuration are in you can begin to test your calibrations. It is not recommended to fly your UAV immediately after this, but to bench test instead. Then the recommended procedure for testing, according to the writers of this code, is to run the stabilize mode first. Then test using the fly-by-wire mode, to make sure that the autopilot will not over correct during flight. Here is a list of the other flight modes offered in this program:

- Manual- Full manual control
- Circle- Fly in a stabilized circle, this is used when there is no GPS present
- Stabilize- This mode will have the plane maintain level flight
- Fly-by-wire A-Autopilot style control via user input, manual throttle
- Fly-by-wire B-Autopilot style control via user input, airspeed controlled throttle
- Auto-All control of the UAV are through the ArduPilot
- RTL-The UAV will return to its launch location and circle until manually controlled
- Loiter- The UAV will circle in the current location

Now that we know how the program needs to be set up lets discuss what is happening under the hood. The ArduPilot is comprised of 27 different tabs that include functions and defined variables that are used to controller the UAV. There are multiple tabs of GPS configurations; these are for the use of different types of GPS modules. There are also configurations, which were discussed in the previous paragraph, for running this autopilot through a simulator. Under the ArduPilot_2_7 tab is where you will find the initialization of the program, there is a main loop that initializes three very important functions, of different speeds, for this program; these functions are loops themselves and are labeled as such: main loop event, medium loop, and fast loop. Below in Figure 9.2.1

is a graphical visualization of the flow of the main loop. The main loop is designed to run as fast as possible and we are looking to reach about 50Hz.

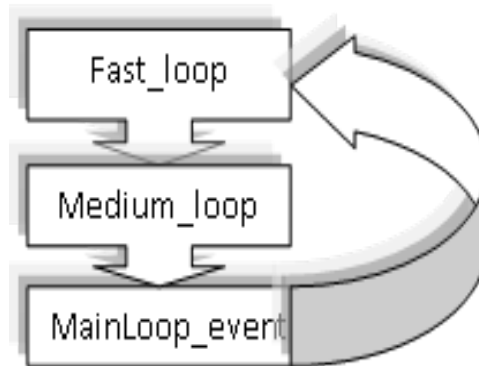


Figure 9.2.1 - Main Loop

In Figure 9.2.1, the main loop begins with the fast loop. The fast loop, as its name implies, is designed to run without any delays. The loop takes care of checking the failsafe if it was enabled in the configuration tab. The GPS is updated, a few error calculations are done and the stabilization of the UAV is performed here as well. Below in Figure 9.2.2 is a flow layout of the functions that are initiated within the fast loop. The black arrows indicate the completion of the previous function or statement and the movement to the next function or statement.

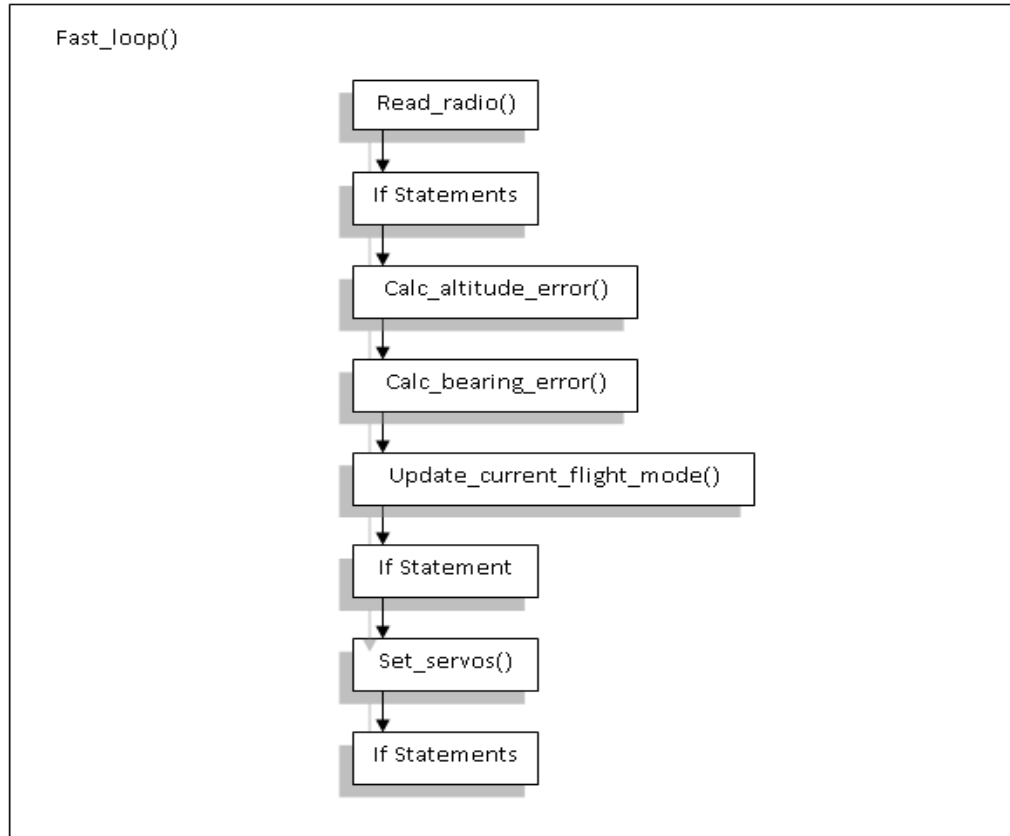


Figure 9.2.2 - Fast Loop Function

Beginning at the top of figure 9.2.2, the first function is the read radio. This function is for reading in the radio transmission from the RC controller. This allows the user full control of the UAV during the auto pilot sequence. The next box has the title of 'If Statements', this is because there is a series of statements that check the configurations that were done in the AP_Config tab. These ranges from checking to see if a GPS is being used or if you are running a simulator, to what type of sensors are being used. The next two functions down are to calculate the error in the altitude and bearing of the UAV to the data of the GPS. Next is the update current flight mode function. This is the function that tells the UAV how to fly based on the flight mode that was selected. These flight modes were listed earlier in the chapter and can be switched using a switch on the RC controller. After the update to flight mode there is a single if statement, can be seen in figure 9.2.2, which just checks to see if the flight mode is greater than manual. What is meant by this, is that the list from earlier is assigned a numerical value beginning with zero. Therefore manual is equal to zero and the statement checks to see if the control mode is greater than zero. If this is true, the statement will execute the stabilize function. The last sets of instructions to be run in the fast loop are set servos and two if statements. The set servos function just writes out the servos PWM values and the two if statements just output performance reports, either to the simulator or to the ground station. After we have completed the fast loop, the main loop will then execute the next function which is the medium loop. This loop is comprised of five 10Hz loop pieces. These loop pieces are called cases and only one case is run during each execution of the medium loop. Below in the Figure 9.2.3 is the flow layout of the medium loop and the cases that are executed.

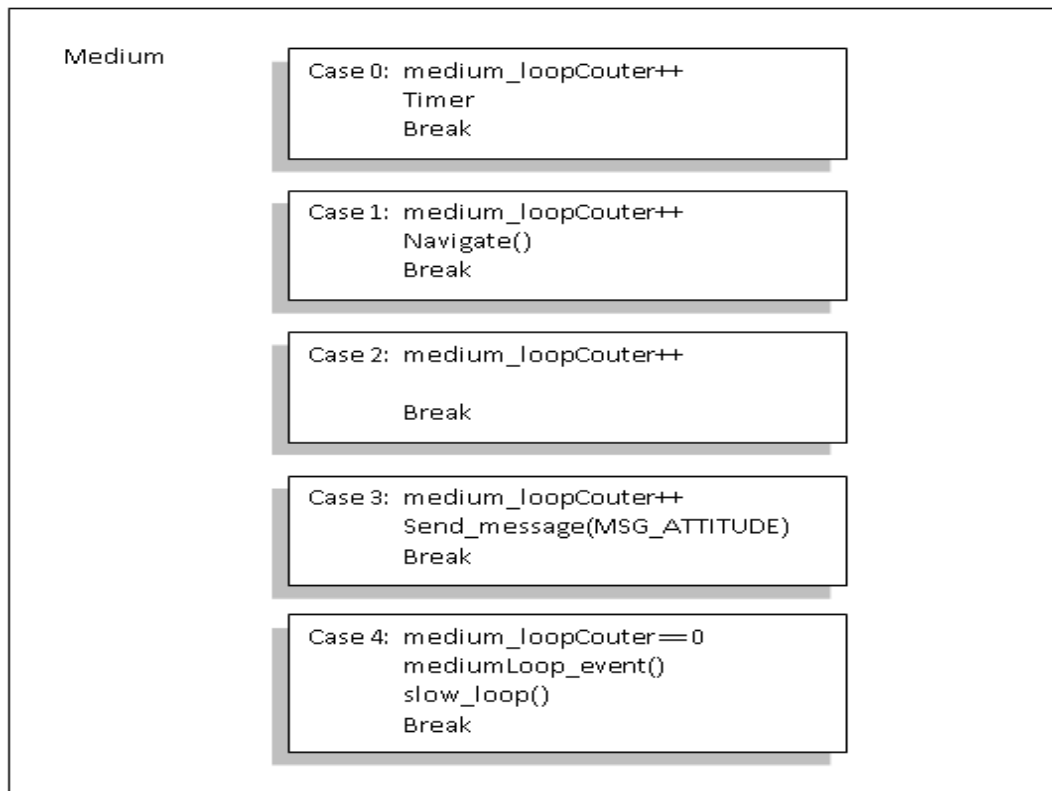


Figure 9.2.3 - Medium Loop Function

Unlike Figure 9.2.2, Figure 9.2.3 does not have any arrows signifying the flow of instructions. This is because when the cases are run and completed, they don't proceed to next case, but instead break the medium loop and return to the main loop. When the medium loop is initially executed, the loop counter is initialized to zero, and then the counter is check to see which case is to be run. So beginning from the counter set to zero; Figure 9.2.3 starts with the case zero. In case zero and all other cases there is a counter increment, so that next time the medium loop is called it will execute the next case. This is not done in case four, the loop counter is reset to zero to restart the medium loop back to the first case when the medium loop is called again.

In case zero there is only one function and that is called update GPS. This function is as it sounds; if a GPS is connected the function will take in the data and decode it to useful information. It checks to make sure that we have a good GPS lock also. The data is collected it is then sent to the ground station. Now that case zero has completed, it will break and return to the main loop. In case one, we also only have one function. This function is called navigate; this function will calculate the UAV's desired bearing. Case two is unused, but this allows us a position in the code to make our modifications. In case three we have a function to send the altitude to the ground station. In case four we have two functions; medium loop event and a slow loop. In the medium loop event, there is no code written. This function is for setting up events such as dropping an egg at a selected

waypoint or take a photo. This also presents us with another place where we could add our code.

The next function we have is the slow loop. The slow loop function is similar to the medium loop function, for it is made up of three different $3\frac{1}{3}$ Hz cases. The first and last cases are empty, allowing the user the opportunity to insert code here. We will not use this section because it runs too slow. This is unusable when checking the altitude of a descending aircraft, where time is of essence. On the other hand, these sections could also be used for up linking. The only case being used, as of current, is case one. In case one, it has three functions; the first of the functions is the read control switch. The switch that is being referenced here, is a 3 position switch located on the radio controller. This switch can be hard code to the user's preference in the configuration tag to switch between flight modes. So in turn, the function checks to see if this has been changed. The next function is read battery. This is only used if the user has selected this option in the configuration tab. For our project we will be using this option; when this function is called it will check the battery voltage and return the information to the ground control station. The last function called is update GPS light; this function will check the GPS lock and update the LEDs accordingly. For example, the lights will blink if the GPS does not have a fix.

After the GPS light has been updated, the case will break and return to the medium loop. Now that the slow loop function was the last function; the last case will break and return to the main loop. Back in the main loop there is one last function that we haven't talked about that would have been called after each case of the medium loop had finish running. This function is the main loop event; it is very similar to the medium loop and slow loop events, in that the user could enter additional code. The difference is the timing of this loop; it will run more frequently which makes it ideal for functions that involve sensor reading or anything that could affect the flight of the UAV. This is why the main loop event function is ideal for the altitude code that we are going to implement. Also, the medium loop could provide an adequate timing for our code. The only way to really determine how often we need to check the altitude will be through physical testing. We will need to see how fast the UAV descends and through that we will determine where to place the altitude gliding code. Below in Figure 9.2.4 is a simple layout of the altitude gliding code. This code may change before the end of the project.

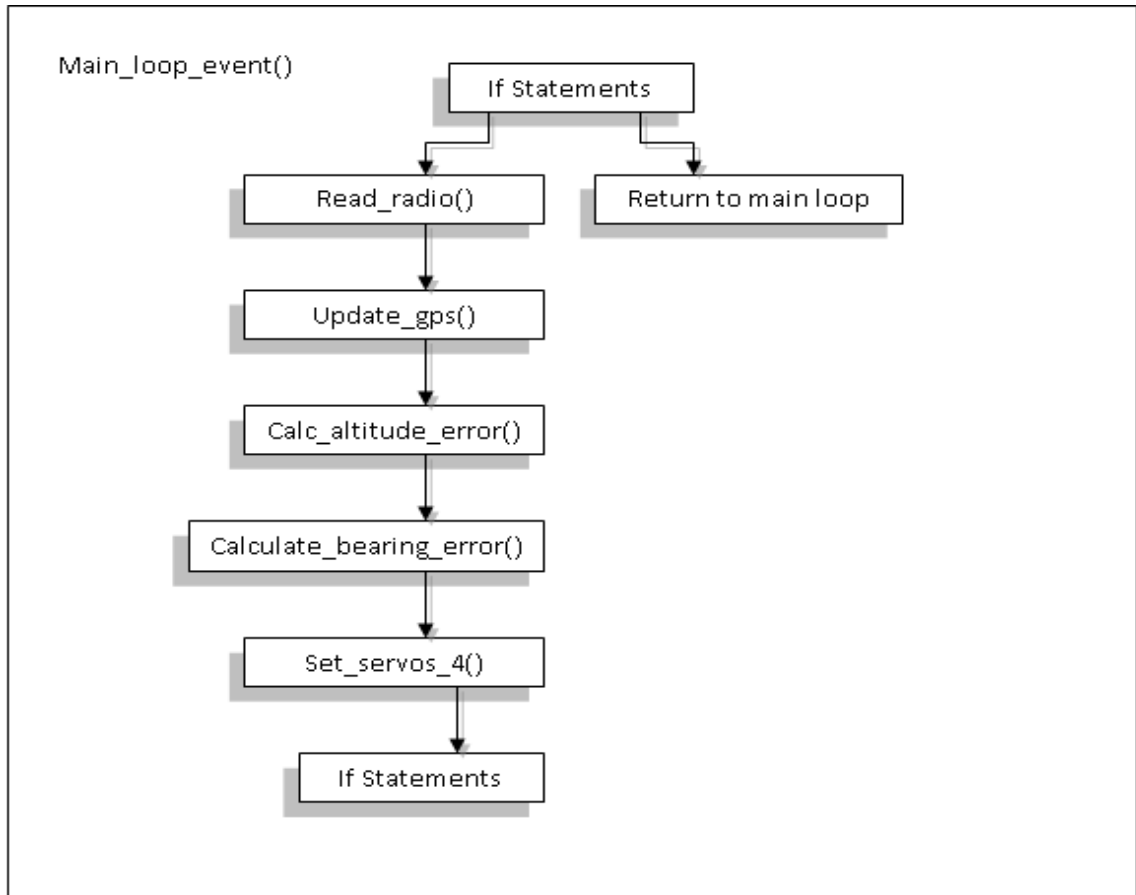


Figure 9.2.4 - The main loop event for allowing the UAV to glide.

First we will be using an ‘If’ statement to check the current altitude against our set minimum altitude. If the altitude is greater than minimum altitude value then the function will end and return to the main loop. But if the altitude was less than the set minimum value, then it would call on the functions required for returning the UAV back to its maximum altitude. First, before returning the UAV to the maximum altitude, it will run the read radio function. It does this to avoid overriding the user, if the user is manually controlling the aircraft. After checking to make sure that the user is not in control; the next function called is the update GPS. This is called to refresh the altitude and location data, so that we are not using old data to adjust the UAV’s altitude. After the update, we run an altitude error check, this is so that we can adjust the servos accordantly; doing this will result in the UAV correcting its altitude. Then we will need to check if our bearing is correct; so we will then need to call calculate bearing error function. This function will check, as it name implies, if the UAV is off course and the then correct its bearing. The next function to be called is set servos function. This function will write out the servo PWM values. Last in our code is a set of ‘if’ statements to check if we are using a simulator or are using a GPS. This is because the program needs to know where to write out the values, into a simulator or to the ground control station. Below in Figure 9.2.5 is a simple representation of what this code is performing.

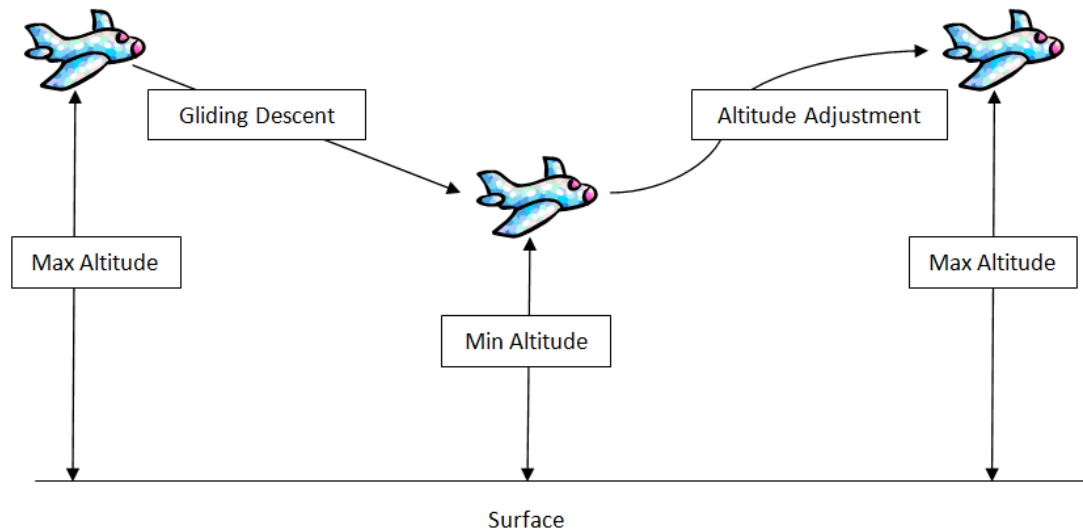


Figure 9.2.5 - Visual representation of the main loop event.

Now that our code has run through, it will return to the main loop. As mentioned before, this code is not permanent. Our code modification could be relocated to a different location within the main code. Our way of determining where to use this modification will be through testing. Initially we will use a simulator to test how the aircraft will be affected from our code modifications; and from there we will be able to correctly place this code or change what functions that need to be used. Once we have the code working how we want it to and if we feel comfortable then we will proceed to testing a test plane.

9.3 Altitude Heading Reference System (AHRS)

For the code above to work correctly, another microcontroller will need to be loaded with the AHRS code. The AHRS code is developed using the Arduino IDE; the code will be used to maintain a model of the UAV's orientation in space. This code is based on Bill Premerlani's Direction Cosine Matrix (DCM) algorithm. The DCM is a 3 by 3 matrix array; the gyro data is used in a time step integration to update the matrix. With this data from the sensors and GPS can be used to correct for errors. These errors are caused by numerical errors. With this data we can determine pitch, roll and yaw; but for this project we will only be using the pitch and yaw. With this code we are now able to use the above coding to fly, without the control of a user, our UAV to waypoints. Plus, with code modification that we have designed, we will be able to extend the flight time of a RC aircraft.

9.4 PC Software

Besides using Arduino for programming the microcontrollers, we will also be using few other programs to help us with the controlling of the UAV. We will be using ArduPilot mission planner, the ground control station and XPlane. The mission planner and the

ground control station are both open source software that can be download for free. For the simulator, there are two options we can choose from. We could use an open source flight simulator called FlightGear. One advantage to using this software is that it is free, but one disadvantage is that the code needs to be hacked to have it communicate properly with ArduPilot code. The second simulator we could use is XPlane. One advantage to using this simulator over the other is that ArduPilot is already configured to use this simulator. The only disadvantage is that it is not open source, so the group would have to purchase a license for thirty dollars to use. Below in the subsection we will discuss the software a bit more.

9.4.1 ArduPilot Mission Planner

The ArduPilot mission Planner is a configuration tool to help you easily plan your UAV's missions. Missions are a set of waypoints for the UAV's flight path. Below in Figure 9.4.1 is the interface of the mission planner. You will notice the mission planner uses Google maps for laying down the waypoints; on the map you can see a home point. This home can be clicked and dragged to any location you require.

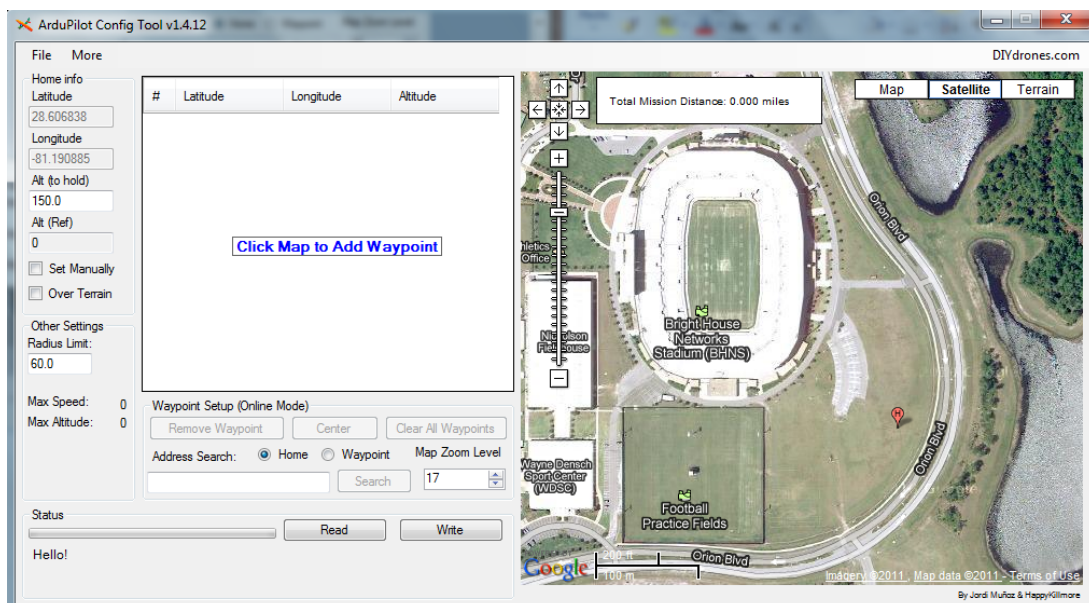


Figure 9.4.1 - Mission Planner with home location at Bright House Stadium

The mission planner is to be used after the ArduPilot has been loaded to your board. It will be loaded through the FTDI cable that was used to load the auto pilot code. Before we power up the board, we need to disconnect the GPS to avoid any serial conflicts. The mission planner supports up to 48 waypoints and those waypoints can be added by clicking on the map. Below in Figure 9.4.2 is a picture of the mission planner with some waypoints added. Below in Figure 9.4.2 you can see that four waypoints were added. The location of those waypoints is now displayed on the map and in the window next to the map. There is a check box off to the left of that window, titled over terrain, will allow the waypoints to be offset by the altitude at that waypoint. If you uncheck this box the UAV

will fly at the fix altitude enter in the box titled “alt to hold”. The altitudes can be display in feet or meters. In Figures 9.4.1 and 9.4.2 the altitudes are displayed in feet. (NOTE: This is not the final the mission that will be carried out for testing.)

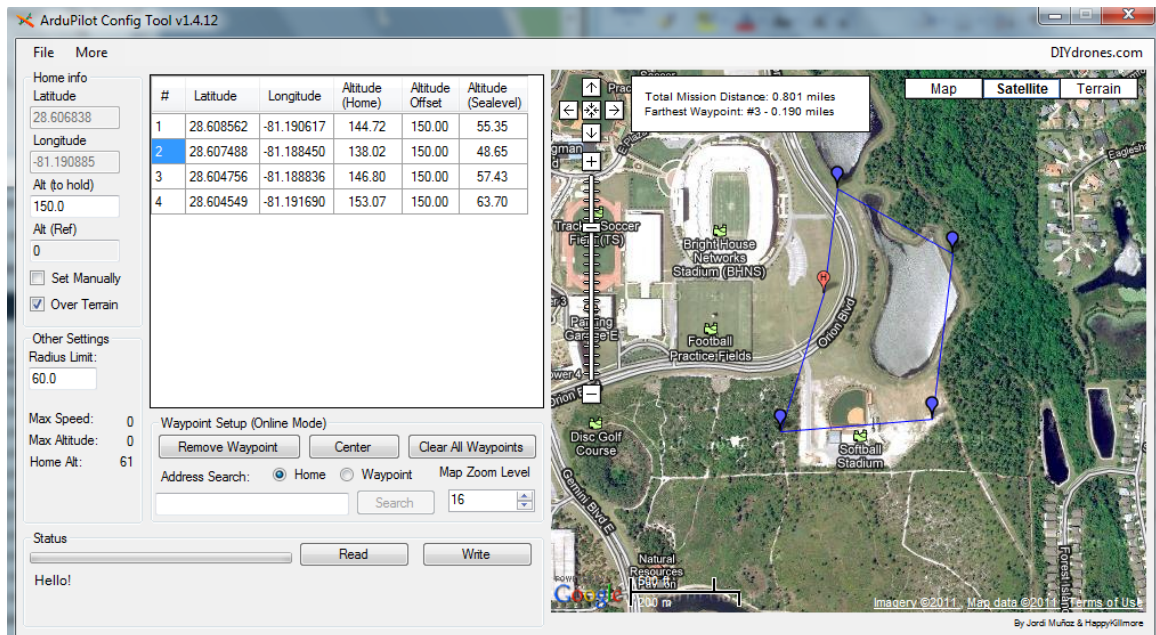


Figure 9.4.2 - Mission Planner with waypoints selected

There are a few other features that can be tweaked before uploading the mission to the microcontroller. You can set the radius of the waypoint in the max radius box. This is how close to the waypoint the UAV needs to get before moving to the next waypoint. Once you have made all the configurations for your mission, you then click the write and it will store this mission to non-volatile memory. When the mission is loaded it does not load the home point; this is loaded when you do the field set up. If you wanted to check to make sure that the mission was correctly loaded then you can click the read button and it will load the waypoints on to the map. The read button also allows you to load the data from a recent flight; you will be able to see the pervious mission, max speed and max altitude. These all can be cleared using the “Clear Max Alt/Speed” in the More menu. Under the More menu, you can also offset the “Roll trim” and “Elevator trim”; this is to compensate for the UAV tendencies to drift. This happens when the UAV’s XY sensors are not level.

9.4.2 Ground Control Station

The ground control station is an optional addition to the project that uses Xbee modules to communicate to the UAV. For this project we have decided to use this feature. Below in Figure 9.4.3 is the interface of the ground control station. As you can see in the right half of the screen the program utilizes Google Earth. During the flight of the UAV the ArduPilot code creates a log of the location and altitude; and that is sent to the ground station using the Xbee modules, where it is updated on Google Earth.

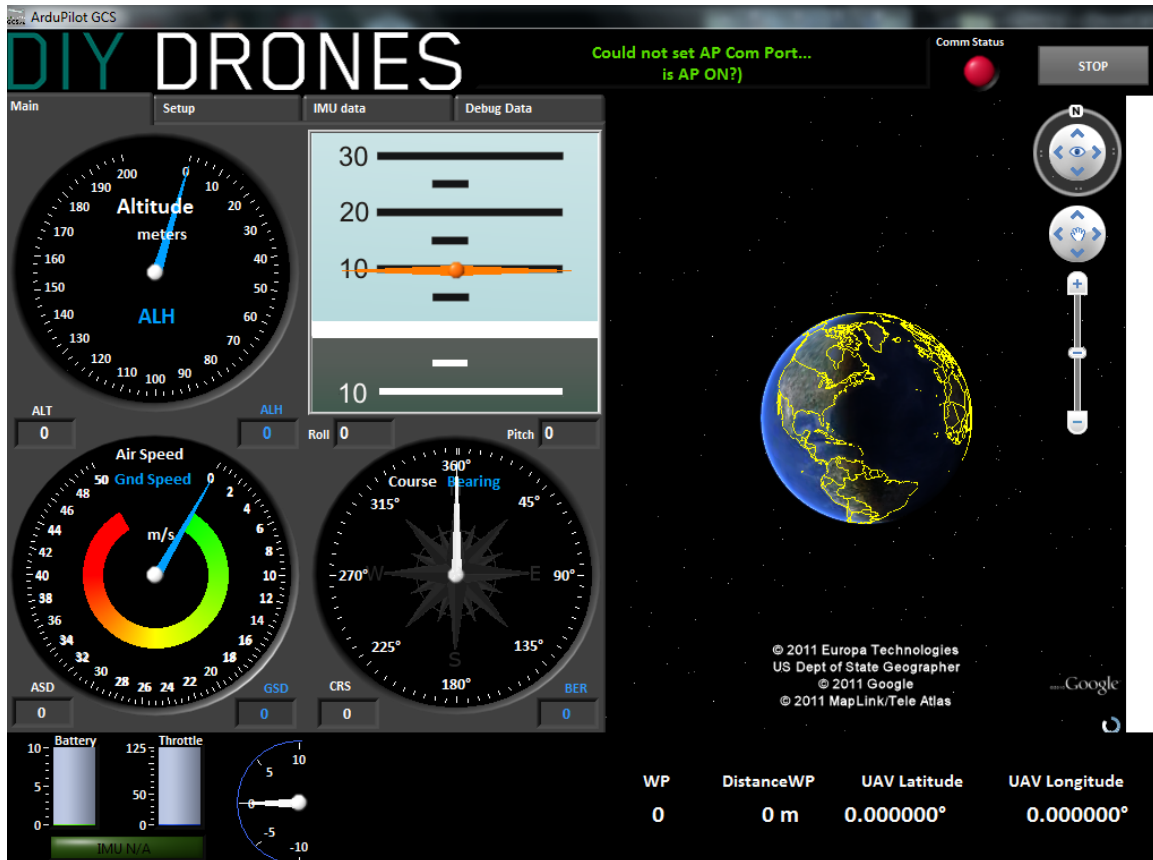


Figure 9.4.4 - Ground Control Station

As you can see, the gages all read zero, this is because we do not have the board connected. Because of this the Google Earth will zoom in to the latitude and longitude coordinates at the bottom of the screen. If there were a board connected and we were flying the UVA the gages would read accordingly. We would see the altitude, the level, and bearing of the UAV during flight. Plus, in the lower left-hand corner we will also be able to monitor the battery and throttle level. For the battery monitoring to work we will need to jump a wire for the positive terminal to board. The only sensor that we might not be using would be the air speed sensor, but this could be added later. For the ground control station to work properly we will need to set some configurations, like the baud rate for you GPS and the COM port for the Xbee. The ground control station also supports audio updates, so the user can keep his eyes on the sky. The ground control station does not have to be only used during the flight of the UAV; it can also be used in conjunction with the simulator that you are using.

9.4.3 Simulator

The XPlane simulator can be used to simulate the ArduPilot code, which is why this is perfect for our project. This will allow us to test the modified code without destroying any model planes. With the board connected and running XPlane we can also practice

flying the plane through the RC controller. We do not need the GPS connected because that information will be produced from XPlane. The only downside to using this software is that we need to purchase a license to use, but in the long run if it saves us from having to buy one than one airframe; I believe that justifies the cost of the program.

Chapter 10: Hardware Design

The autopilot design considered to be the heart and the brain of the SEV. The design itself is all about making the SEV function effectively by bringing all the devices together. Table 10.1 is a reminder of all the equipment which is going to be used to build the autopilot system. The devices were mentioned earlier in this paper; however, for the purpose of putting everything together they are restated in Table 10.1.

The Devices	Their Purposes
Barometric Pressure Sensor	It senses the elevation due to the change in the atmospheric pressure with respect to ground
Triple-axis Accelerometer	It senses the status of the SEV with respect to angular measurements
Dual-axis Gyroscope	It senses the status of the SEV with respect to the X and Y axes
Single-axis Gyroscope	It senses the status of the SEV with respect to the Z axis
GPS	It tells the position of the SEV
Two Microcontrollers	They receive all the information above and process them as programmed to control the motor and the servos
Two Servos	The microcontroller tells them how they should behave to keep the SEV flying safely
Brushless Motor	The microcontroller tells it how it should behave during the flight
Zigbee	It transmits the data from the microcontroller to the ground control station and receives data from it as well

Table 10.1 - Autopilot System

Wiring the autopilot system is very important for the system to function as expected; the embedded devices should be either mounted on the circuit board or wired, both serve the same purpose of making the system function efficiently. However, the system is first going to be integrated on a UAV development platform for the purpose of programming the microcontrollers as well as for testing the functionality of the overall system. Once that is already done, it is time to put the equipment together to show the hardware design as it appears in Figure 10.1. The two microcontrollers appear to be reserved for navigation and autopilot controlling. The Navigation ATmega328 receives data from the barometric pressure sensor, GPS, accelerometer, and the two gyroscopes to report them to the autopilot microcontroller which is going to communicate with the ground

controlling unit via the RF module. Sometimes based on the received data the user interrupts the hardcoded program for the autopilot to save the SEV.

The user is able to send specific orders regarding the throttle, rudder, and the elevator to the pilot controller via Zigbee which is attached to the pilot controller. In short, Zigbee is being used to transmit and receive data to and from the ground controlling unit. The RC receiver is an additional tool to the system that is being used in case of an emergency to control the SEV in flying, landing, and from being out of range if the program, unfortunately, happened to fail. The pilot controlling unit is programmed already to take a full care of the SEV for the full flight; it responds to the data being received from the navigation controller to increase or decrease the throttle as well as controlling the elevator depending on the elevation of the SEV. In addition, depending on its stability the rudder, elevator, and throttle have to response to keep the SEV from losing its path.

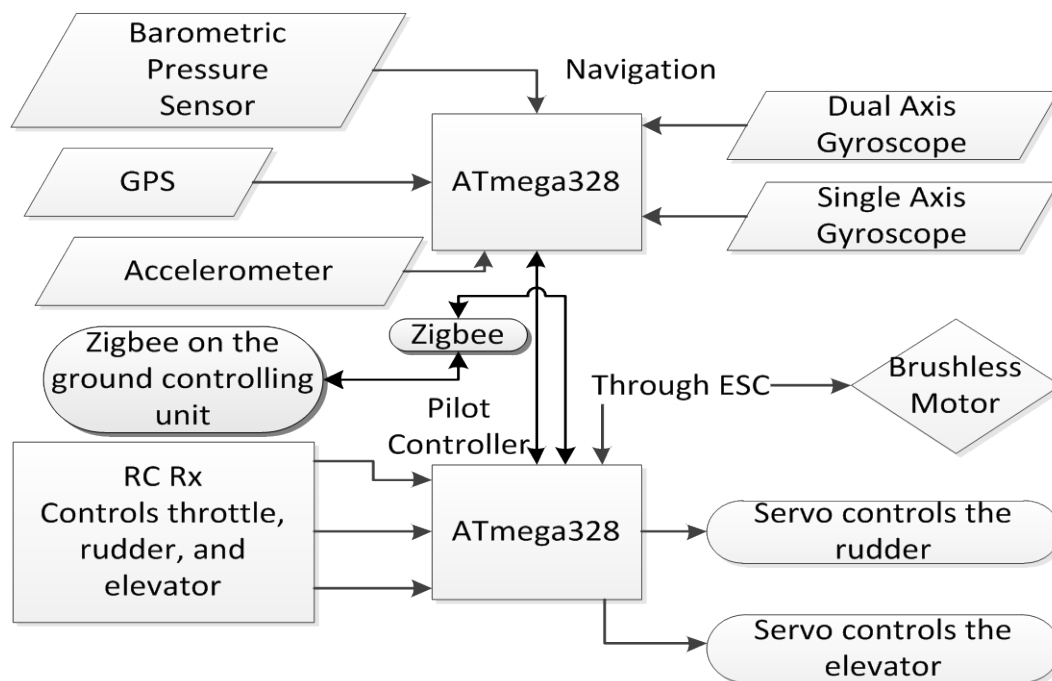


Figure 10.1 - Hardware Design

As it appears in the previous figure, the hardware design consists of having two microcontrollers. Each controller will be handled separately in greater details before they are connected. However, before Figure 10.2 which is the pin configurations figure, Table 10.2 comes first to uncover most of the pin descriptions which serves as an introduction to the wiring process of the system as well as introducing Figure 10.2. The contents of Table 10.2 are according to Atmel's user manual of ATmega328 which explains all the pins in greater details. Ports PC0 to PC3 which serve as the analog inputs to the A/D converter is going to be used to connect the ATmega328 with the other components that output analog signals. However, Ports PD0, PD1, PC4, and PC5 are used as serial inputs or outputs to connect the digital devices with the ATmega328. Table 10.2 also describes in more details other pins named as V_{CC} , GND, Reset', XTAL1, XTAL2, AV_{CC} , and

AREF. In addition, the table mentions specific connections that will help in installing the devices to avoid some errors such as connecting the AV_{CC} to the V_{CC} even if the analog to digital converter is not used as well as connecting the GND of the A/D converter to ground. The table also specifies how many seconds the reset pin takes to reset the device which will never happen if the reset pin was not set on a low level.

Pin Descriptions	
V_{CC} (digital supply voltage)	1.8 – 5.5 V
GND (digital ground)	0 V
Port B (PB5...PB0)	Port B is an 8-bit bi-directional I/O port with internal pull-up resistors. It is used for specific functions
Port C (PC5...PC0)	Port C is an 8-bit bi-directional I/O port with internal pull-up resistors. It is also used for specific functions
Port D(PD7...PD0)	Port D is an 8-bit bi-directional I/O port with internal pull-up resistors. It used for specific functions as well
PC6/Reset' (reset input)	If the RSTDISBL is not programmed PC6 is used as a reset input. A low level on this pin for longer than the minimum pulse width will generate a Reset, even if the clock is not running
PB6/XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit
PB7/XTAL2	Out from the inverting Oscillator amplifier
AV_{CC}	It is the supply voltage pin for the A/D converter, PC3-PC0 and ADC7-ADC6. It should be externally connected to V_{CC} , even if the A/D is not used. If the ADC is used, it should be connected to V_{CC} through a low pass filter. PC6-PC4 use digital supply voltage, V_{CC}
AREF	It is the analog reference pin for the A/D Converter.
ADC7:6	In the TQFP package, ADC7:6 serve as analog inputs to the A/D converter. They are powered by AV_{CC} and serve as 10-bits ADC channels

Table 10.2 - Pin Descriptions according to Atmel's user manual

All the previous pin descriptions will help understand the next figure which is figure 10.2 to represent the pin configurations of the device. Figure 10.2 is going to speak for itself; it is captured from Atmel's user manual to show the pin configurations of the ATmega328. The purpose of presenting the pin configurations of the device is to help understand the nature of the chosen microcontrollers and to introduce how each microcontroller will be handled during the course of the hardware design of this project. The dimensions of the ATmega328 are 7mm in width and 7mm in length. Each microcontroller comes with 32 pins total, 8 pins along each side of the ATmega328. Once the microcontrollers are programmed and ready to be mounted on the circuit board, the other components come into the picture to represent how the SEV is functioning. All the

devices will be either mounted on the circuit board or wired to the circuit board to make the system function as represented in the hardware design. Two ATmega328 microcontrollers will be installed in the project for the SEV to fly in a given range that is specified by two given altitudes; in other words, the microcontrollers will be programmed as it is explained in the software design chapter of this project. In short, the hardcoded ATmega328 takes all the received information from all the different sensors to send them to the pilot controller unit which is able to analyze the received data to deliver a clear message to the servos and the motor for the purpose of making the SEV either glide or elevate. In addition, the pilot microcontroller is also able to communicate with the ground controlling unit.

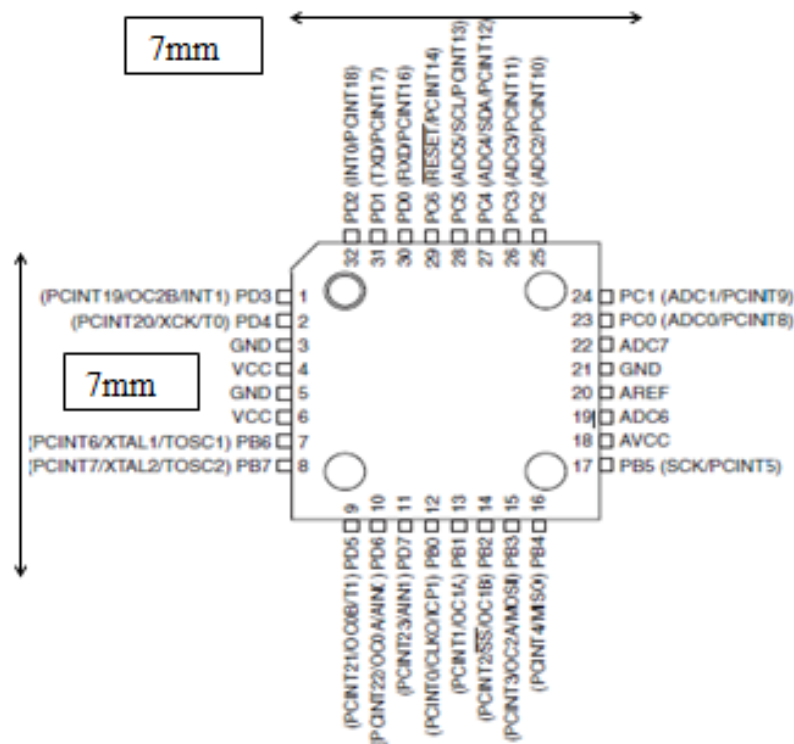


Figure 10.2 – Atmega328 TQFP Pin Configurations.
“Reprinted with the permission of Atmel”

Since the top view of the ATmega328 was presented already to show its pin configurations, now it is time to present all the components to put them together with respect to their pin configurations and functionalities. However, it is better to bring all the components in the picture as in introduction to the next step of the hardware design of the project. Therefore, Table 10.3 is presented to divide the components into two categories to help in the wiring process of the system. The two categories are analog components and digital components. The drive behind breaking down the components into two categories is to plug each device into the right pin on the ATmega328. The table contains the following components: Triple-Axis Accelerometer(MMA7361L), Dual-Axis Gyroscope (X, Y)(IDG-500), Single-Axis Gyroscope (Z)(ISZ-500), Barometric Pressure Sensor(BMP085), CopernicusTM GPS receiver, ATmega328 microcontroller, and Zigbee.

The mentioned components represent the navigation function of the hardware design which was shown in figure 10.1. The accelerometer and the two gyroscopes are analog components. They send their data as analog signals to the navigator microcontroller through the analog to digital converter pins, which are PC0 to PC3 as well as ADC6 and ADC7. On the other hand, the GPS, pilot microcontroller ATmega328, and the Zigbee are digital components; except for the ATmega328 which is able to transmit and receive both digital and analog. The digital components are going to be connected through the serial ports such as PD0, PD1, PC4 and PC5.

Part Name (Number)	Analog Device	Digital Device
Triple-Axis Accelerometer (MMA7361L)	√	
Dual-Axis Gyroscope (IDG-500)	√	
Single-Axis Gyroscope (ISZ-500)	√	
Barometric Pressure Sensor (BMP085)		√
Copernicus™ GPS		√
ATmega328	√	√
Zigbee		√

Table 10.3 - Components Analysis

The navigation unit of the design is solely presented in figure 10.3 to narrow down the discussion on the sensors, the GPS and the communication method of the design. The navigation function is the brain of the SEV. It receives all the signals and sends them to the pilot controller to process them in order to respond appropriately. The two microcontrollers will communicate to form a complete Autopilot system. The data will be sent to the pilot controller which sends signals to adjust the servos, and controls the motor via the ESC. It also reports all the data to the ground station via Xbee. The top section of the figure is placed on the right hand side of the printed circuit board.

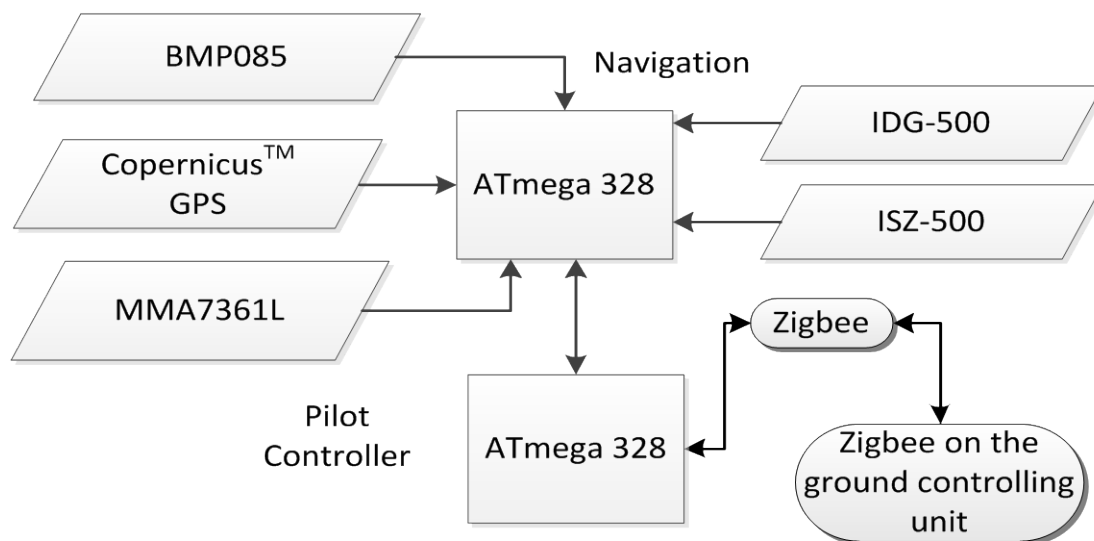


Figure 10.3 - Navigation Unit

It is important to follow some order in presenting the material; therefore, the same order of Table 10.3 will be followed to avoid any kind of misunderstanding. The Triple-Axis Accelerometer (MMA7361) is going to be presented first, respectively followed by the Dual-Axis Gyroscope (IDG-500), the Single-Axis Gyroscope (ISZ-5000), the Barometric Pressure Sensor (BMP085), the Copernicus™ GPS receiver, the second microcontroller, and finally the RF module. For the design to be completed, the pin descriptions and the pin configurations of the accelerometer have to be presented first to achieve the aimed design. The pin descriptions of the MMA7361L, according to the Freescale Semiconductor technical data sheet, are presented in Table 10.4. The specified device comes as 3mm in width and 5mm in length; in that compact space it has 14 pins. Three pins are used as analog outputs which are X_{OUT} , Y_{OUT} , and Z_{OUT} . Their pin numbers are 2, 3 and 4 respectively. The power supply input of the device, pin number 6, is typically 3.3V. Its maximum voltage input is 3.6V, and its minimum voltage input is 2.2V. The device also provides a sleep mode which is ideal for the SEV. When the sleep mode is active, the device shuts down the three axis outputs which reduces the operating current. A low input signal on pin number 7 will place the device on the sleep mode and will reduce the current to $3\mu A$.

Number	Pin	Description
1	N/C	No internal connection leave unconnected
2	X_{OUT}	X direction output voltage
3	Y_{OUT}	Y direction output voltage
4	Z_{OUT}	Z direction output voltage
5	V_{SS}	Power supply ground
6	V_{DD}	Power supply input
7	Sleep'	Logic input pin to enable product or sleep mode
8	N/C	No internal connection leave unconnected
9	0g-Detect	Linear free-fall digital logic output signal
10	g-Select	Logic input pin to select g level
11	N/C	Unused for factory trim leave unconnected
12	N/C	Unused for factory trim leave unconnected
13	Self-Test	Input pin to initiate Self-Test
14	N/C	Unused for factory trim leave unconnected

Table 10.4 - MMA7361L Pin Descriptions

Table 10.4 was presented to introduce the pin configurations of the device. Six pins are aligned to be on the left side of the device and six pins on the right side of it. Two pins left; each pin located in front of the other. One pin is on the upper side of the device, and the other pin is on the lower side. Figure 10.4 will show the previous descriptions of the shape, size, and the location of the pins.

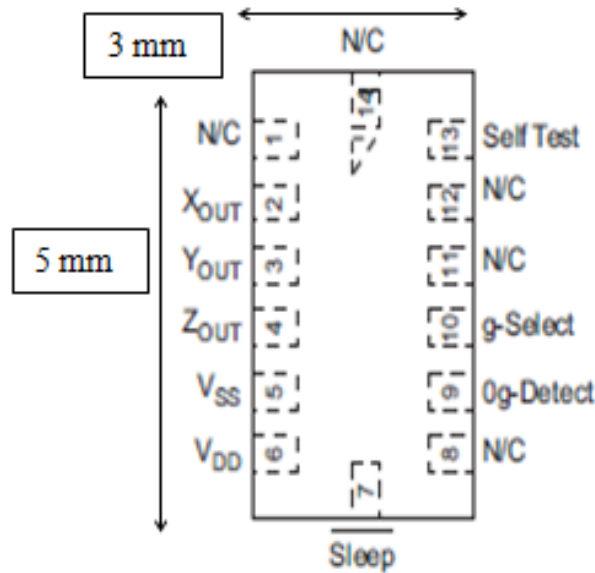


Figure 10.4 – MMA7361L Pin Configurations.
 “Reprinted with the permission of Freescale”

Since the pin descriptions and configurations were introduced already, it is time to connect the accelerometer to the microcontroller. The Chosen microcontroller, ATmega328, has six analog to digital converter pins, PC0, PC1, PC2, PC3, ADC6, and ADC7, which give the opportunity to connect the triple axis accelerometer to three of the mentioned pins. Table 10.5 shows the appropriate wiring to connect the accelerometer to the microcontroller. Pin number 2, X_{OUT} , on the MMA7361L is connected to the analog to digital converter port PC0; Pin number 3, Y_{OUT} , on the MMA7361L is connected to port PC1; Pin number 4, Z_{OUT} , on the MMA7361L is connected to PC2. Pin number 7 on the MMA7361 which enables the sleep mode is connected to PB0, and finally the 0g-detect is connected to PD2 to interrupt the system if the MMA7361L detects a free fall of the design.

MMA7361L	ATmega328
X_{OUT} (Pin No.2)	PC0 (A/D _{IN} Pin No.23)
Y_{OUT} (Pin No.3)	PC1 (A/D _{IN} Pin No.24)
Z_{OUT} (Pin No.4)	PC2 (A/D _{IN} Pin No.25)
Sleep' (Pin No.7)	PB0 (CLKO Pin No.12)
0g-Detect (Pin No.9)	PD2 (Pin No.32)

Table 10.5 - MMA7361L Connections

It is also shown in Figure 10.5 how the accelerometer is interfaced with an arbitrary microcontroller as suggested by Freescale. This connection will be taken into consideration to avoid any errors. It is also specified by Freescale the values on the capacitors, which appear in Figure 10.5. According to MMA7361L technical data sheet, it is recommended to “use 0.1 μ F capacitor on V_{DD} to decouple the power source; in

addition, the physical coupling distance of the accelerometer to the microcontroller should be minimal. It also specifies to use 3.3nF capacitor on the outputs of the accelerometer to minimize clock noise”.

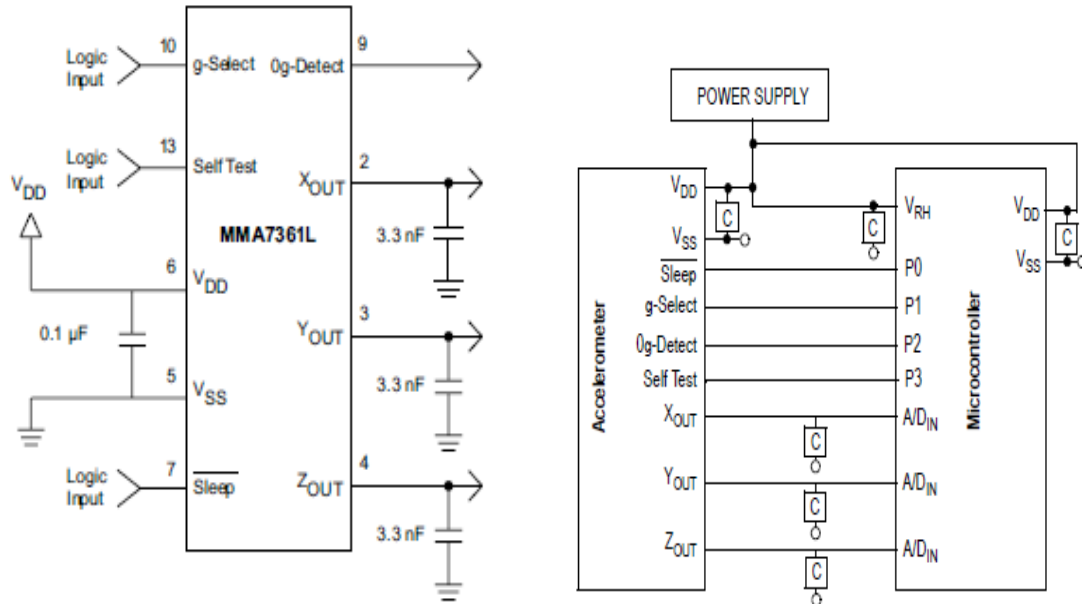


Figure 10.5 - Recommended interfacing accelerometer to microcontroller.
“Reprinted with the permission of Freescale”

Next, the Dual-Axis Gyroscope is about to be brought into the picture. For the design to be completed, the pin descriptions and the pin configurations have to be presented to achieve the aimed design. The pin descriptions of the IDG-500, according to the InvenSense specifications sheet, are presented in Table 10.6. The specified device comes as 4 mm in length and 5 mm in width; in that compact space it has 28 pins. There are four pins are used as analog outputs. Two outputs represent the X_{OUT} and $X_{4.5OUT}$; two outputs represent the Y_{OUT} and $Y_{4.5OUT}$ as well. The rest of the pins will be explained in the table and in the next figure for better illustrations. However, both pin numbers 9 and 19 which are the positive supply voltage are typically powered by 3V. Its minimum operating power supply voltage is 2.7V, and its maximum operating power supply voltage is 3.3V. Two power supply inputs on the device which are pins number 9 and 19; both pins have to be connected for the device to operate properly. Pin number 24 is the auto zero function which is used to maximize the gyro’s dynamic range when using pins number 7 and 14 as the outputs. The auto zero function works by keeping the zero rate output as close to the voltage reference, 1.35V, as possible to give the user of the gyroscope a wider signal range. On the other hand, If the auto zero function is not used, it has to be connected to ground.

Number	Pin	Description
2, 8, 26, 27, 28	GND	Ground
9, 19	V_{DD}	Positive Supply Voltage

1	X _{OUT}	Rate output for rotation about the X-axis
5	X _{4.5IN}	X-axis input to the 4.5X amplifier
6	XAGC	Amplitude control capacitor connection
7	X _{4.5OUT}	X-axis output of the 4.5X amplifier
12	CP _{OUT}	Charge pump capacitor connection
14	Y _{4.5OUT}	Y-axis output of the 4.5X amplifier
15	YAGC	Amplitude control capacitor connection
16	Y _{4.5IN}	Y-axis input to the 4.5X amplifier
20	Y _{OUT}	Rate output for rotation about the Y-axis
22	V _{REF}	Precession reference output
23	PTATS	Temperature sensor output
24	AZ	X and Y Auto Zero control pin
10, 11, 13, 21, 25	RESV	Reserved. Do not connect
3, 4, 17, 18	N/C	Not internally connected

Table 10.6 - Pin Descriptions of the IDG-500

Table 10.6 was presented to introduce the pin configurations of the device in figure 10.6. In the figure, the dimensions are presented to illustrate how the gyroscope is going to be connected to the microcontroller. It is also specified by the manufacture, InvenSense, that the two axes the device rotate about have to be aligned with the axes of the SEV to get appropriate rotational rates. Figure 10.6 is presented with the permission of Invensense; the figure is captured from the IDG-500 product specifications to enhance the presentation of the design. As shown in the figure, whenever the device rotates about either the X or Y axis, it produces a positive output voltage. This positive output signal tells the microcontroller whether the SEV is elevating about the X-axis or rolling about the Y-axis. If there was no signal detected at the output, the SEV must be flying in level and not rolling. It is might be possible for the SEV to fly in level, but it must be difficult no to roll due to the presence of the winds and the absence of the ailerons. Thus, the outputs are expected to send signals constantly.

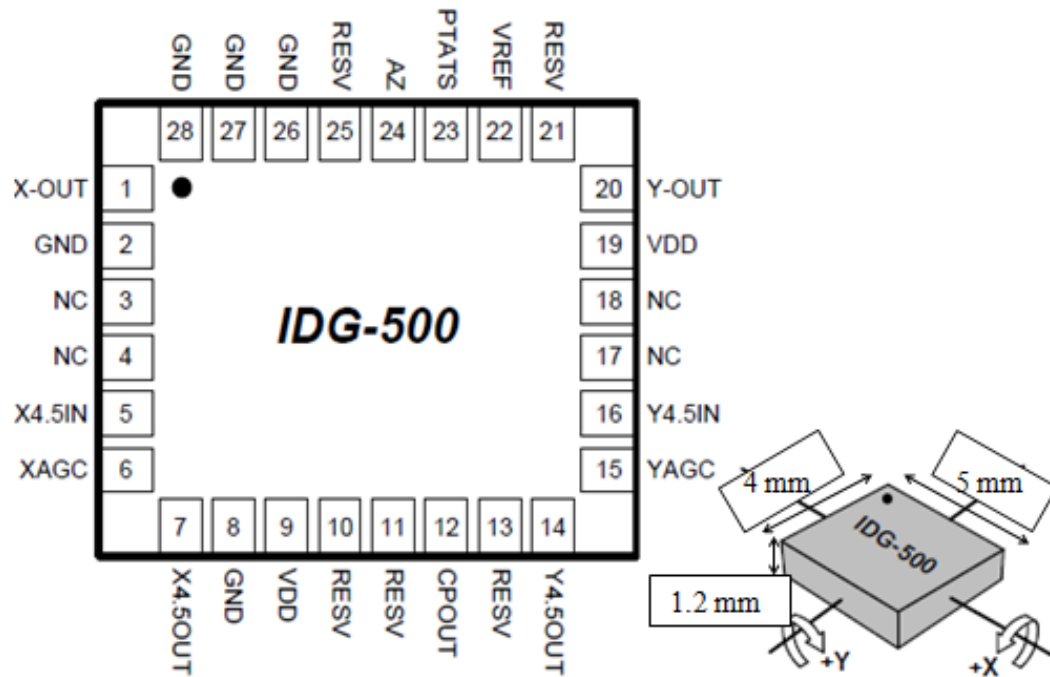


Figure 10.6 – IDG-500 Pin Configurations and Axes Representation.
 “Reprinted with the permission of InvenSense” (pending)

Since the pin descriptions and configurations were introduced already, it is time to connect the dual axis gyroscope to the microcontroller. The Chosen microcontroller, ATmega328, has six analog to digital converter pins which gives the opportunity to connect the IDG-500 to two of the mentioned pins. Table 10.7 shows the appropriate wiring to connect the dual axis gyroscope to the microcontroller. The 4.5outputs are chosen for both axes because they give two plus bits of resolution when the output signal is digitized. Meanwhile pins number 19 and 22, analog to digital inputs on the microcontroller, which are ADC6 and ADC7 respectively serve as 10-bits analog to digital converter channels.

IDG-500	ATmega328
X _{4.5OUT} (Pin No.7)	ADC6 (A/D _{IN} Pin No.19)
Y _{4.5OUT} (Pin No.17)	ADC7 (A/D _{IN} Pin No.22)

Table 10.7 - IDG-500 Connection

In addition to the previous information, figure 10.7, which is the reference circuit, shows how the dual axis gyroscope should be connected with respect to its external circuitry; the suggested connections by InvenSense shall be taken into consideration while integrating the gyroscope with the microcontroller. The provided technical information will be used to avoid any mistakes. It is also specified by InvenSense the values on the capacitors and resistors which make low pass filters on the output as it appears in the figure below. The low pass filter capacitors are 0.1 μ F; the low pass filter resistors are 750 Ω .

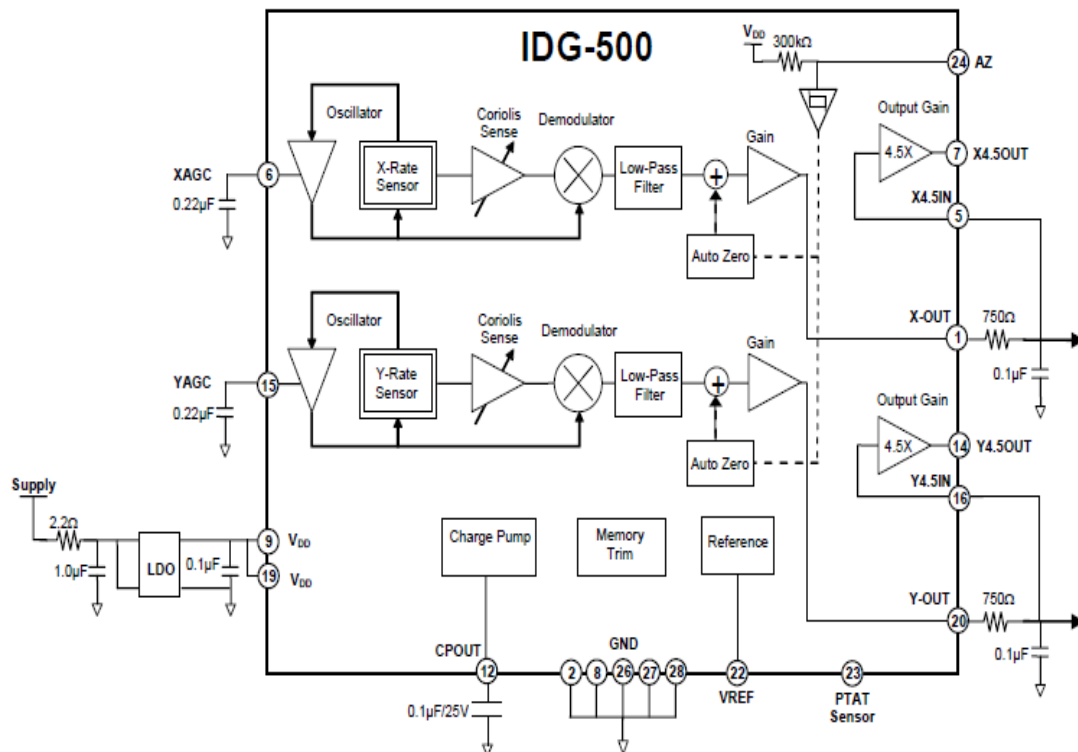


Figure 10.7 - IDG-500 Reference Circuit.

“Reprinted with the permission of InvenSense” (pending)

The design is still missing a single axis gyroscope to sense the rotational rate about the Z-axis. A common design argument occurred over choosing a separate gyroscope to complete the rotational sensing ability of the SEV and why not using a triple axis gyroscope. However, the reason behind selecting two different gyroscopes does not hide behind the functionality of the available gyroscopes; on the other hand, the reason is closely related to the software design of the project. The software design of having a triple axis gyroscope simply gets very complicated when it is actually compared to installing two different gyroscopes. Thus, The ISZ-500 gyroscope which is made by InvenSense was chosen to complete the rotational sensing about the Z-axis, the perpendicular axis to the surface of the gyroscope. Since it is just another gyroscope made by the same company that functions exactly the same as the dual axis gyroscope, however, on a smaller scale, there is no point of representing the previous material once again. The same pin descriptions, pin configurations, and the same dimensions as well. However, the sensing rate is about the Z-axis. It comes with two analog outputs Z_{OUT} and $Z_{4.5OUT}$.

Since the pin descriptions and configurations were introduced previously, it is time to connect the single axis gyroscope to the microcontroller. The Chosen microcontroller, ATmega328, has six analog to digital converter pins which gives the opportunity to connect the MMA7361L to PC0, PC1, and PC2. In addition, it also gives the opportunity to connect the IDG-500 to ADC6 and ADC7; finally, the ATmega has only one of the six

analog outputs left to connect ISZ-500. Table 10.8 shows the appropriate wiring to connect the previously mentioned components to the microcontroller. The 4.5 outputs were chosen for both axes of the IDG-500, but the regular Z output is chosen for the ISZ-500 due to the limited 10-bit pins on the microcontroller which functions analog to digital channels.

ATmega328	ISZ-500	IDG-500	MMA7361L
PC0(A/D _{IN} Pin No.23)	-	-	X _{OUT} (Pin No.2)
PC1(A/D _{IN} Pin No.24)	-	-	Y _{OUT} (Pin No.3)
PC2(A/D _{IN} Pin No.25)	-	-	Z _{OUT} (Pin No.4)
PC3(A/D _{IN} Pin No.26)	Z _{OUT} (Pin No.20)	-	-
PD2(Pin No.32)	-	-	0g-Detect(Pin No.9)
ADC6(A/D _{IN} Pin No.19)	-	X _{4.5OUT} (Pin No.7)	-
ADC7(A/D _{IN} Pin No.22)	-	Y _{4.5OUT} (Pin No.17)	-

Table 10.8 - ISZ-500, IDG-500, and MMA7361L Connection

The only sensor that is missing from getting all the sensors connected to the microcontroller is the digitized barometric pressure sensor, BMP085, which produced by BOSCH. The pin descriptions of the BMP085, according to its specification sheet by BOSCH, are to be visited for more clarifications about the functionality of the digital pressure sensor. The specified device comes as 5 mm in width and 5 mm in length; in that compact space it has eight pins. Its advantage is that it could be easily integrated with the microcontroller due to the I²C interface. The I²C interface require only two bus lines: a serial data line (SDA), and a serial clock line (SCL). Once the device is connected to the microcontroller by the I²C bus, there is going to be a software relationship between them. The BMP085 is going to be the slave machine and the ATmega328 is going to be the master receiver machine to collect all the data. The I²C reads calibration data from the EEPROM as soon as the ADC is finished.

As shown in Figure 10.8, 7 pins that are wired which indicate that there is an internally not connected pin which is pin number 5. Pin number 1 is the ground; pin number 2 is the end of conversion (EOC) digital output; pin number 3 is the power supply V_{DDA}; pin number 4 is the digital power supply V_{DDD}; pin number 6 is the I²C serial bus clock input (SCL); pin number 7 is the I²C serial bi-directional data bus (SDA); finally, pin number 8 is the master clear (active low) input (XCLR). The power supply ranges between 1.8v and 3.6v; the external resistors are typically 4.7K Ω ; however, the value of the resistors ranges between 2.2k Ω and 10k Ω . Figure 10.8 not only represents the pressure sensor, but it integrates it with an arbitrary microcontroller which is going to be considered carefully while designing this project. This integration focuses on connecting the pins with the microcontroller as well as how the BMP085 process the data through the ADC and EEPROM to the control unit.

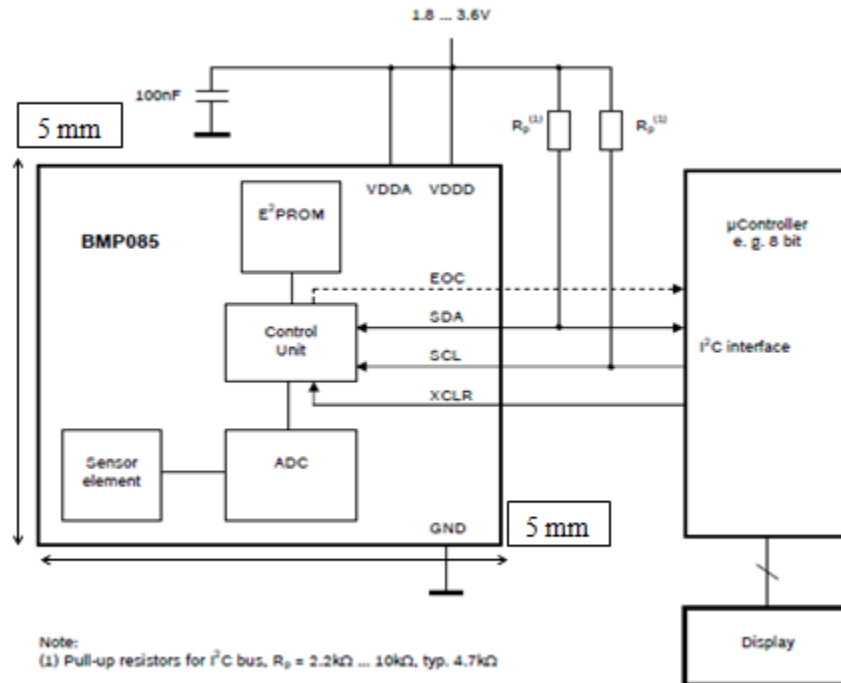


Figure 10.8 - BMP085 integration.
“Reprinted with the permission of BOSCH”

Table 10.9 shows how the pressure sensor is going to be integrated with the ATmega328 pin by pin. This very important step is really the resultant of the previous presentation of the pressure sensor. It all comes to this table. The connections of the SCL and SDA with PC5 and PC4 respectively should be connected as shown in the previous figure to be able to make other connections on the two-wire interface lines on the ATmega328 microcontroller.

BMP085	ATmega328
Serial Bus Clock Input (SCL Pin No.6)	PC5 (Pin No.28)
Serial Bus Data Bi-directional Line (SDA Pin No.7)	PC4 (Pin No. 27)

Table 10.9 - BMP085 Pin connections

Currently the design is missing multiple devices on the navigator hardware design unit. The missing components are the GPS, Zigbee, and the master microcontroller which is the pilot microcontroller. The components are going to be visited in the following order: GPS which is to be connected to the navigation microcontroller, pilot microcontroller which is to be connected to the navigation microcontroller as well, and Zigbee which is to be connected to the pilot microcontroller. The CopernicusTM GPS receiver, which is produced by Trimble, comes in a 19 mm in width and 19 mm in length. When it is powered, in the range of $2.7 V_{DC}$ to $3.3 V_{DC}$, all the twenty eight pins are able to function. Thus, According to Copernicus user manual by Trimble, Table 10.10 is

presented to introduce the pin descriptions of the Copernicus™ GPS receiver. Hint, “O” stands for output; “I” stands for input; “G” stands for ground; “P” stands for power.

Number	Pin	Description
1, 13,14, 15, 27, 28	GND (G)	Signal ground. Connect to common ground
2, 4	GND (G)	One of two RF grounds. Connect to RF ground system
3	RF Input (I)	50Ω unbalanced (coaxial) RF input
5	LNA_XEN (O)	Can be used with active antennas only. Active low logic level to control external LNA
6, 17, 18, 22, 25, 26	Reserved (I/O)	Do not connect
7, 8	Open, short (I)	Logic level from external detection circuit
9, 10	Reserved (I)	Connect to V _{CC}
11	XRESET (I)	Active low logic level reset. Connect to V _{CC}
12	VCC (P)	Supply Voltage
16	XSTANDBY (I)	Selects “RUN” or “STANDBY” mode. Connect to V _{CC} if not used (run only)
19	PPS (O)	Logic level timing signal at 1 Hz. Do not connect if not used
20	RXD_B (I)	Logic level secondary serial port receiver
21	RXD_A (I)	Logic level primary serial port receiver
23	TXD_A (O)	Logic level primary serial port transmitter
24	TXD_B (O)	Logic level secondary serial port transmitter

Table 10.10 - the GPS Pin Descriptions

Based on the previous table, one of the most important pins is the RF input, pin number 3, which is explicitly required to be connected to a coaxial antenna to detect the satellites signal. Therefore, a compact passive coaxial antenna is going to be attached to the GPS through pin number 3. Fortunately, the Copernicus™ GPS includes a built-in low noise amplifier to overcome the coax cable losses. Passive antennas usually are within 3 feet from the GPS which indicates the no need to use an external low noise amplifier, pin number 5 LNA_XEN. On the other hand, an active antenna that is more than 3 feet away from the GPS does require the external low noise amplifier to overcome the cable losses. Figure 10.9 presents the device and shows its passive antenna-minimum connections according to its user manual by Trimble. It appears that the open and the short, pins number 7 and 8 respectively, are kept unconnected; thus, the Copernicus reports an open antenna condition which is equivalent to setting both pins to 1 if using an active antenna. The use of pins number 7 and 8 is really to monitor the status of the external low noise amplifier which is not needed for this design; hence, the open and short pins are kept floating. The reset, pin number 11, is connected to the V_{CC}, and it is only active when it is held low for 100 microseconds. Since figure represents the minimum connections of the GPS, it is shown in the figure that the standby, pin number 16 which controls the run and standby conditions, is connected to the V_{CC}, pin number 12, to indicate that it is running. Now, according to Copernicus user manual by Trimble, the pin configurations of the GPS

are to be presented to visualize how the antenna is connected and how the GPS is going to be integrated with the navigation microcontroller.

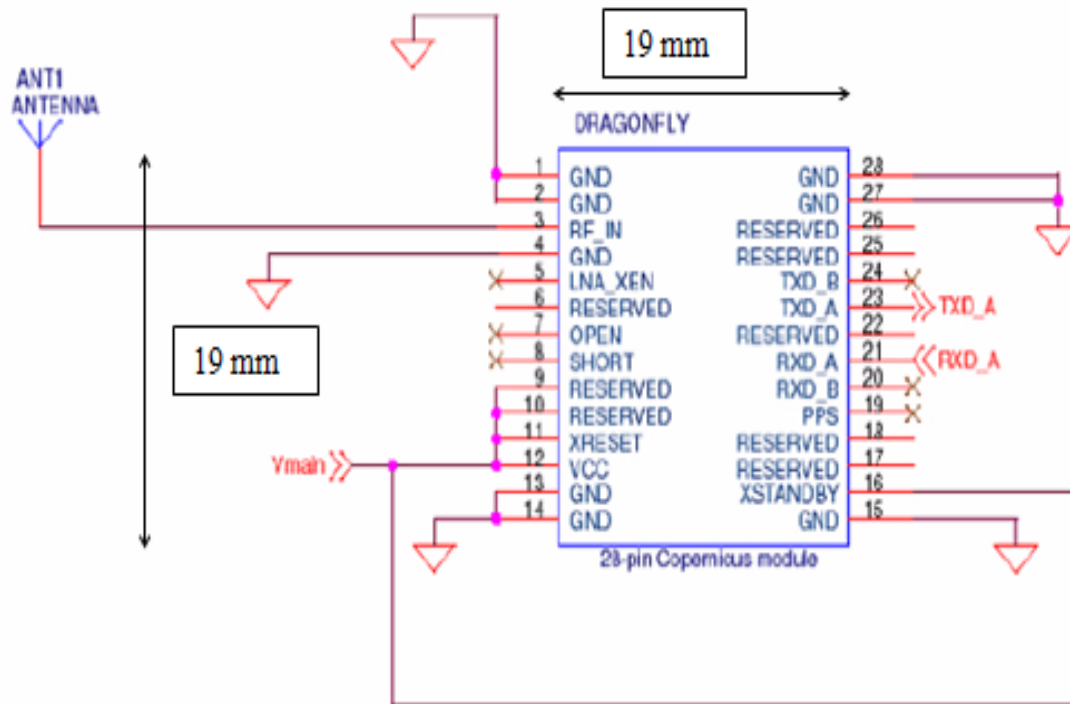


Figure 10.9 - Copernicus™ Pin Configurations and minimum Antenna Connections.
“Reprinted with the permission of Trimble”

Table 10.11 shows how the Copernicus GPS receiver is going to be integrated with the ATmega328. This very important step is really the resultant of the previous presentation of Table 10.10 and Figure 10.9. The connections with the microcontroller is presented in Table 10.11. The connections of the TXD_A, pin number 23, and RXD_A, pin number 21, which are two serial ports to transmit and receive data into and from the microcontroller through PD0, USART input pin, and PD1, USART output pin.

Copernicus GPS Receiver	ATmega328
TXD_A (Pin No.23)	PD0 (RXD Pin No.30)
RXD_A (Pin No.21)	PD1 (TXD Pin No.31)

Table 10.11 - Copernicus™ GPS Connections

The navigation microcontroller unit is almost done except for connecting it with the master microcontroller which is the pilot microcontroller unit. Actually, currently the brain of the SEV which collects all the data is complete; however, the heart which keeps the SEV alive; in other words keeps the SEV from falling is not yet connected to the brain to accomplish the overall design of the system. The pilot microcontroller is the master microcontroller which receives all the collected data at the navigation ATmega328. It also receives the RC signals at 2.4 GHz which are sent to the ATmega328

when an emergency occurs. This microcontroller communicates with the ground controlling unit via Zigbee; it also controls the servos and the throttle on the brushless motor via the electronic speed controller. The previous description of the pilot controller unit is presented in figure 10.10. This microcontroller is programmed not to exceed the maximum allowed altitude which is set by the Federal Aviation Administration. Their rules and regulations explicitly prohibit flying beyond an altitude of 400 feet. Therefore, the master microcontroller will be set to fly at a maximum altitude of 400 feet and a minimum altitude of 200 feet. If the SEV reaches the 400 feet, the master ATmega328 turns off the motor and starts to glide; however, once the SEV gets to the lower end, the master ATmega328 does exactly the opposite. It turns on the motor and starts to elevate. It is also programmed to receive orders from the user at ground controlling unit which is able to override the written code.

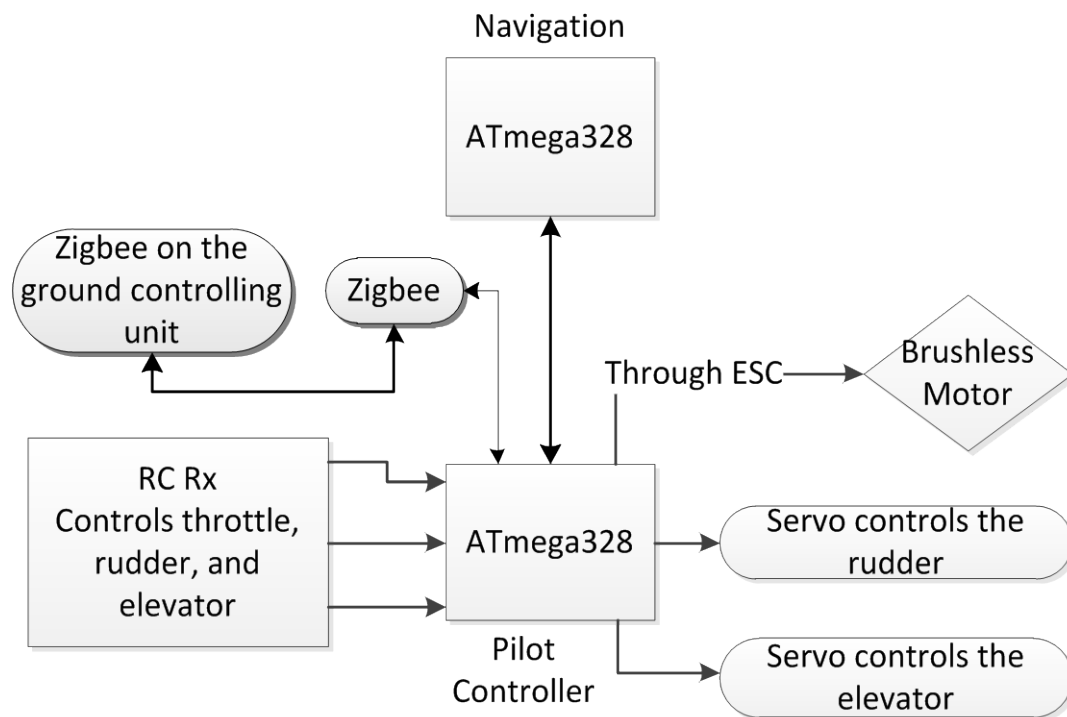


Figure 10.10 - the Pilot Controller Unit

Based on the preceding figure, the design requires the slave microcontroller to communicate with the master microcontroller by sending and receiving data; this connection allows the brain and the heart of the project to function in parallel. This connection is reached via the two wire interface. The slave microcontroller is already connected to the barometric pressure sensor via the I²C; nevertheless, the two wire interface allows for multiple device connections. This double connection on the two wire interface is shown in Figure 10.11.

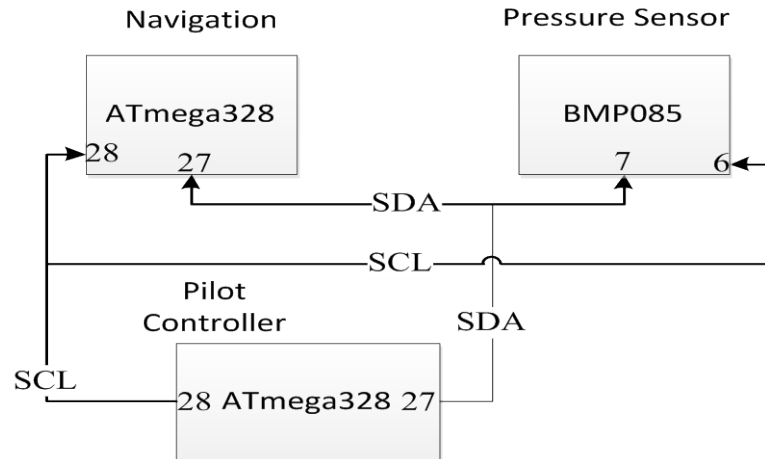


Figure 10.11 - the Pilot Controller Integration

The Pilot controller, ATmega328, now is required to communicate with the ground station. The communication between the microcontroller and the ground station is managed by Zigbee Pro 900. The design is going to have two RF modules to transmit and receive data from the SEV. Zigbee is going to handle all the collected data by the sensors and the GPS to report them to the ground controlling unit. In order to do so the ATmega328 needs to be connected to Zigbee to transmit the data to the RF module. Then the RF module is going to detect the ground station Zigbee and transmit them at a frequency of 900 MHz. Once the RF module on the ground stations receives the data, it encrypts them to show them on the screen to the user. If the user needs to react based on what was received, the same process happens to tell the ATmega328 on the project to perform in a specific manner. Figure 10.12 shows the full picture regarding the communication process of the system. On the ground station, the connection between the device and the laptop is going to be through a Universal Serial Bus (USB) wire, and the connection between the RF module and the microcontroller on the SEV is going to be via the Universal Asynchronous serial Receiver and Transmitter (UART) module connections.

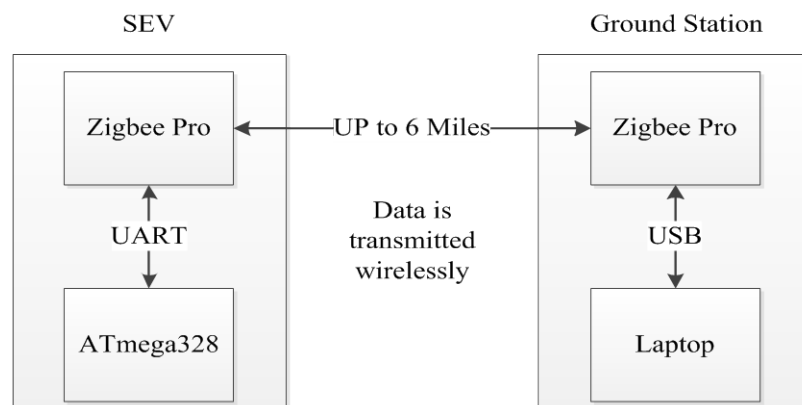


Figure 10.12 - Data Communication

For an accurate connection between the microcontroller and the RF module it is important to make sure that the microcontroller is able to communicate by the UART. Actually, the ATmega328 supports a Universal Synchronous and Asynchronous serial Receiver and Transmit (USART) connections which is very flexible serial communication device. On the ATmega328 side some work has to be done to be able to connect it with the Zigbee Pro. The USART consists of three major blocks: a Clock Generator, Transmitter, and Receiver. The generator clock does support four modes of clock operation: normal asynchronous, double speed asynchronous, master synchronous and double slave synchronous mode. According to Atmel's user guide of the ATmega328, "The universal mode select (UMSELn) bit in USART Control and Status Register C (UCSRnC) selects between asynchronous and synchronous operation. Double speed (asynchronous mode only) is controlled by the U2Xn found in UCSRnA register." Therefore, to be able to set the USART on the UART mode the UMSELn bit on the UCSRnC register to zero; otherwise, it is the synchronous mode.

Since the microcontroller and the RF module are ready to be connected, the pin descriptions of the Zigbee Pro are to be first presented in table 10.12, then the pin configurations will be presented in a figure to show its size and where the pins located to prepare for the connection with the ATmega328. Zigbee Pro is a transceiver module which is typically powered by 3.3V. The current input needed when data transmitted is 180mA; however, the current need once it receives data is 80mA, both true once the supply voltage is 3.3V. Its operating frequency band starts at 902 MHz and terminates at 928MHz. The RF module is 24.38 mm in width and 32.94 in length; it comes with 20 pins; however, only the minimum connections pins, V_{CC} , D_{OUT} , D_{IN} , GND, are going to be presented in Table 10.12.

Number	Pin	Description	Max Voltage
1	V_{CC}	Power Supply	3.6V
2	D_{OUT}	UART Data Out	V_{CC}
3	D_{IN}	UART Data In	5V
10	GND	Ground	0V

Table 10.12 - Zigbee Pro pin Descriptions

According to the previous minimum connection table, the important pins are 1, 2, 3, and 10. Figure 10.13, According to Zigbee-Pro 900user manual by Digi, shows the internal flow diagram of the module which shows the processor and both the transmitter and the receiver. It also shows the RF switch and the antenna port. On the far left side of the figure, it shows the data input (D_{IN}), voltage supply (V_{CC}), ground (GND), and the data output (D_{OUT}) pin. The serial receiver buffer receives the data coming from pin number 3 (D_{IN}); the data stays inside the buffer until they can be processed. There are some times the serial receiver buffer cannot process the data immediately; for instance, if large amount of data are sent to the module, the CTS flow control may be required to avoid overflowing the serial receive buffer. Serial transmit buffer acts upon receiving an RF data which then sends them out of port number 2 (D_{OUT}). If the RF data packet which is being sent through the transmit buffer does not fit the serial transmit buffer, it is then

dropped. The CTS and RTS both control the flow of the serial data. The CTS stops the serial data from being sent to the module; the RTS stops the module from sending to the serial transmit buffer out the UART.

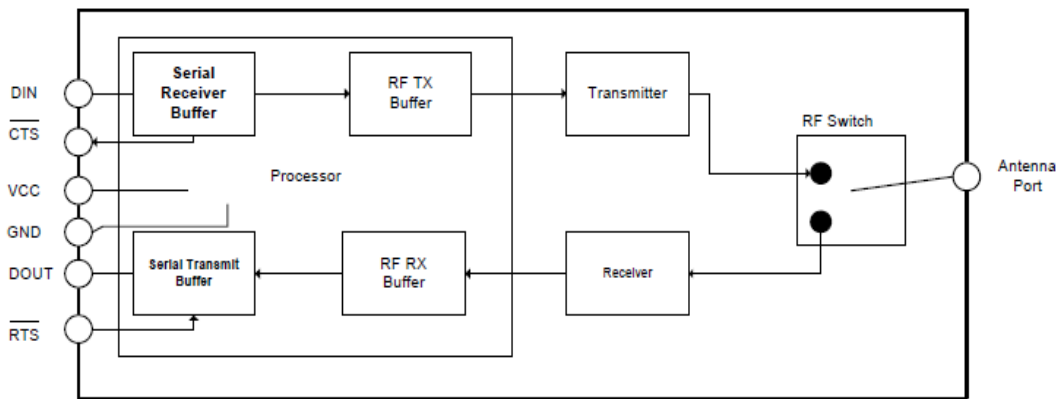


Figure 10.13 - The Internal Flow Diagram.
“Reprinted with the permission of Digi”

Table 10.13 shows how the RF module is going to be integrated with the ATmega328 on the SEV as well as it shows the connection on the ground station. This table of connection connects both devices via their UART ports to perform what was being explained in the previous figure. Pins number 30 (RXD) and 31 (TXD) on the microcontroller are the two serial ports which designated to receive and transmit as USART ports. The RXD port on the microcontroller is going to be connected to port number 2 on the RF module, and the TXD port on the microcontroller is going to be connected to port number 3 on the RF module.

Zigbee-Pro 900	ATmega328
D _{OUT} (Pin No.2)	PD0 (RXD Pin No.30)
D _{IN} (Pin No.3)	PD1 (TXD Pin No.31)

Table 10.13 - Zigbee-Pro Connections

Chapter 11: Design Summary of Hardware and Software

The airframe research section relates to the initial research steps that were taken shortly after our project was chosen. This research began by selecting a plane to use as the frame of our S.E.V. design. We started by looking at RC planes and then later decided to choose an airframe that more closely resembled a UAV. Once we had chosen a hollow frame, we needed to research motors that would be applicable to our project. The motor would be one of the most significant components of the design since it would generate the power for the entire system. Next, we needed to find servos that could be used for the minor mechanical functions. The last component is the electronic speed controller. The ESC would enable the motor and servos to be controlled from a single source. The ESC would

serve as the link between the remote control and the plane. All of these components operate as the core system of the plane. When the project was started, a great deal of time went into researching an appropriate frame for our Solar Extended Vehicle. We decided that the vehicle should be a plane which greatly limited the type of equipment that could be added to our vehicle. Additionally, this put weight limits on all of the extra electronics. Also, the amount of space that would be available was limited by this decision. As a result, the selection was narrowed down to meet all the previous requirements. Therefore, the following airframes are the ones that qualify: Apprentice 15E, Parkzone Radian, and the ProJet RQ-11.

Since the ProJet RQ-11 has several distinct features that separate it from other planes, it is the chosen because it has an advantage regarding its weight and capacity. The light weight and portable nature of the Raven makes it an ideal candidate for multiple operations. Furthermore, the ability to attach additional devices such as cameras and other sensors is another important benefit of this particular plane. The rear propeller is also notable because the placement of the motor in the center of the plane provides an ideal center of gravity. This allows us to add weight evenly around the plane without shifting the overall weight ratio and disrupting the aerodynamics. Another significant part of the airframe of the Raven is the overhead wing. An overhead wing is ideal for gliding with minimal power from the motor. Also, the bend in the main wing enables the plane to maneuver easily without the need for ailerons. The control of the pitch, roll, and turning of the aircraft will be performed exclusively from the rear of the plane. Due to the simplicity of the plane's flying controls, the Raven can easily be adapted to fly completely autonomously. In addition to remote controlled flying, the design will also have the ability to fly autonomously via GPS. The absence of ailerons on the fixed-wing of the aircraft allows the Raven to be flown smoothly without the need for human-guided precision. Figure 11.1 Shows below is a picture of the ProJet RQ-11.



Figure 11.1 - ProJet RQ-11
"Reprinted with the permission of Nitroplanes"

Once the airframe had been chosen, the next step was to find a suitable motor. The ProJet RQ-11 can accommodate a wide range of motors effectively. However, we need to select one that was the best fit for our particular design. The S.E.V. project has unique features that are new to the ProJet frame. Due to these innovative changes, the motor was chosen to be the Power 10 Brushless Outrunner Motor. Therefore, we listed below the specifications of the chosen motor.

Model #	EFLM4010A
Size:	10-size
RPM / Volt:	1100Kv
Max Voltage	7.2-12 V
Input Watts:	
Resistance:	.043 ohms
Idle Current:	2.10 A @ 10V
Continuous Current:	32 A
Max Burst Current:	42A (15 sec)
Cells Ni-Cd/Ni-MH:	6-20
Cells Li-Po:	2-3
Weight:	122 g (4.3 oz)
Overall Diameter:	35mm (1.40 in)
Shaft Diameter:	5mm (.20 in)
Overall Length:	43mm (1.60 in)
Type:	Brushless out-runner

Table 11.1 – Motor Specs

The SEV also requires two servos in addition to the motor. While the motor is connected to the propeller, two servos are needed to control the rear rudder and elevator. The rudder and elevator are located on the tail end of the airframe. The rudder runs vertically with the tail fin while the elevator is connected on a horizontal hinge below the rudder. These two flaps will be the main method of maneuvering the plane while it is controlled via remote control or by GPS navigation. The T-Pro MG90S servo is the chosen one to control the rudder and elevator; hence, two are needed to control both. Therefore, we listed below the specifications of the chosen servo.

Model	T-Pro MG90S
Dimensions	23 x 12.2 x 29mm
Weight	9g (0.32oz)
Operating Speed (4.8V)	0.11sec/60degree
Torque (4.8V)	2.0kg/cm
Operating Voltage	4.8-6.0V

Table 11.2 – Servo Specs

The Electronic Speed Controller or ESC is the third major electronic component that is necessary to allow the plane to fly effectively. The ESC will work in sync with the motor to provide smooth acceleration during flight. Basically, the ESC is what regulates the throttle to the motor. Therefore, the electronics portion of our design is required to communicate with the ESC to control the throttle with respect to the information provided by the data collective devices. The Selection of the speed controller would be very dependent on the specifications of the motor that is used. Since there are multiple choices for applicable motors, we needed to research appropriate speed controllers that would work efficiently with each motor. We found that the ESC that was best compatible with our design is the Platinum-40A-PRO Brushless ESC V1. Therefore, we listed its specifications below.

Model	Platinum-40A-PRO Brushless ESC V1
Cont. Current	40A
Burst Current	60A
BEC Output	5V/3A
Input Voltage	2-6 cells (LiPo)
Weight	38g
Dimensions	59x 27x12mm

Table 11.3 – ESC Specs

The plane is based on a 3-channel system which integrates the motor and two servos. The motor will be the primary source of power for the SEV while the two servos' main functions will be to control the rear rudder and elevator. The electronic speed controller serves as a connection between the radio control at a 2.4 GHz frequency and the motor as well as the servos. The main function of the ESC is to work as a throttle to make precise changes in the plane's airspeed. The microcontroller communicates with the ESC to either increase or decrease the throttle. It also communicates with the servos which thereby translates the received commands to the rudder and elevator.

Therefore, the user is able to maneuver the plane by shifting the alignment of the rudder and elevator. Since the plane does not have ailerons on the main wing, the orientation of the rear rudder and elevator will be the primary method of turning the plane during flight. To summarize, the remote control communicates with the receiver that is hooked up to the microcontroller to send instruction to the servos as well as to the ESC which communicates with the motor. As a result, the user is given full control of both the speed of the plane as well as the pitch and roll to change directions during flight. Below in Figure 10.10 is a visual representation of how these parts are connected.

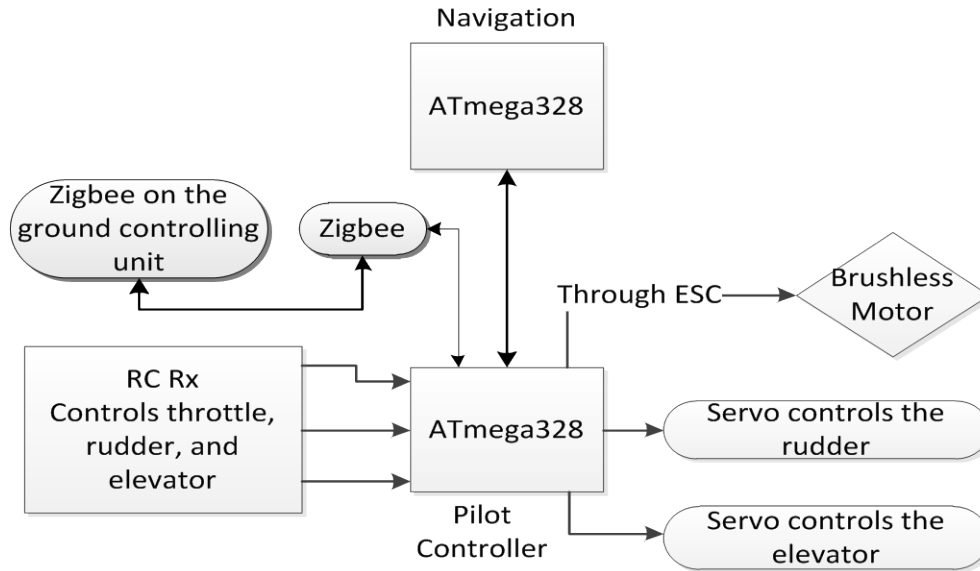


Figure 10.10 - the Pilot Controller Unit

All of the electronics will need power, so they will be run off of a battery mount in the UAV. With all these electronics consuming power the battery will not last for every long. For our UAV project, we decided to use flexible thin cell solar panels that will be attached to the upper wing structure. The allowable area of the wing that we could use to mount our solar cell onto had approximately 525 area inches of space. During our initial project investigation, it was obvious that we had a lot different brands and styles that we could choose to use in our project.

In our preliminary design considerations, we feel that the PowerFilm RC7.2-75 is our best option for charging the battery during flight. The advantage that we have with using the PowerFilm RC7.2-75 is the total weight of the array. Using the PowerFilm RC7.2-75 our total array weighs only 2 ounces. A clear advantage with using the PowerFilm RC7.2-75 is the total thickness of only 0.2mm. This will prove to be very important when we are trying to keep the UAV as aerodynamic as possible. Below in Figure 8.2.2.3 is a possible lay out of the array of cells for the project.

Solar Cells RC7.2-75 100 mA 7.2 V	Solar Cells RC7.2-75 100 mA 7.2 V	Solar Cells RC7.2-75 100 mA 7.2 V	Solar Cells RC7.2-75 100 mA 7.2 V	Solar Cells RC7.2-75 100 mA 7.2 V	Solar Cells RC7.2-75 100 mA 7.2 V
Solar Cells RC7.2-75 100 mA 7.2 V	Solar Cells RC7.2-75 100 mA 7.2 V	Solar Cells RC7.2-75 100 mA 7.2 V	Solar Cells RC7.2-75 100 mA 7.2 V		

**Solar Cells
RC7.2-75 100 mA 7.2 V**

Figure 8.2.2.3 - RC7.2-75

To maximize the total output of the PowerFilm RC7.2-75 solar panels we will have to arrange them in a series/parallel circuit. This is being done by having two groups of four cells and one group of two cells. As stated previously the total power output of the array will be approximately 7.2 watts with a total resistance of 28.8Ω . With each of the cells being rated at 7.2 volts each we will need to have them in series/ parallel connection to accommodate for the 19.8 volts needed to charge our battery system. The minimum wire size that we will use for our project will be 24 gauge copper wire. At the end of the circuit, there will also be a reverse blocking diode such as a 1N5817 to prevent any back feed from the batteries to the solar panels. By using a blocking diode at the end of the circuit, it will also prevent any ground fault in the system thus potentially saving the solar cells from any residual feedback from the system.

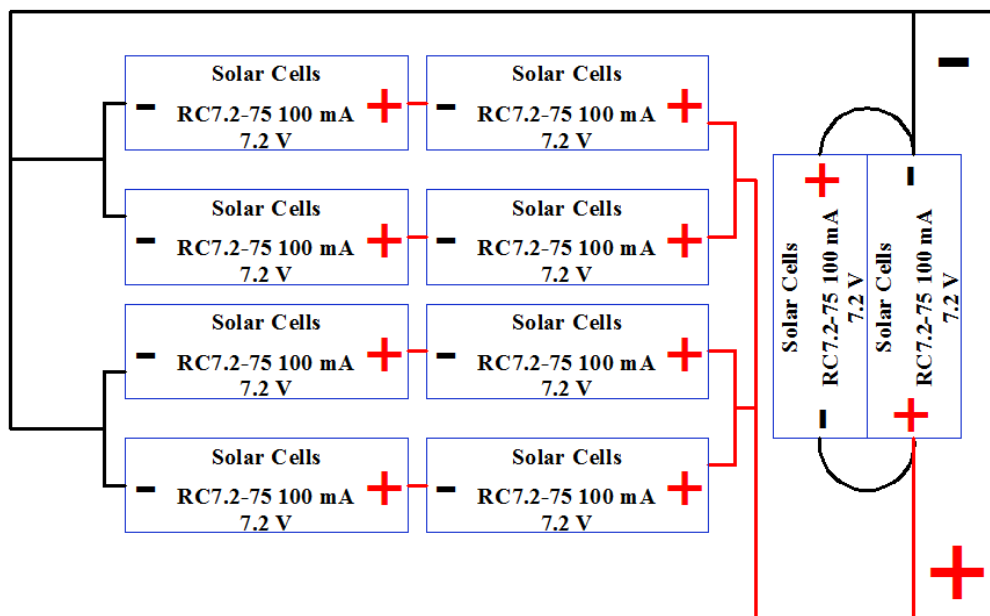


Figure 8.2.3.2 - Solar Array Configuration

The battery pack that we have chosen for our UAV had to match with the correct voltage input of the electric motor that we selected. During our preliminary design criteria, we selected an E-flite EFLB1040. The E-flite EFLB1040 has 2.10 amp current draw at 10 volts when the motor is at an idle speed. The max current draw that we will ever see from the EFLB1040 is 32 amps. Therefore, it was imperative that we chose the correct battery pack size and the correct electronic speed controller. The battery pack that we chose for our project is an E-flite High-Power Series Lithium Polymer EFLB1040 battery. This particular battery pack is commercially available and is very common with RC enthusiast. Most of all it is a perfect match for the E-flite EFLB1040 electric motor that we chose for our project. The E-flite EFLB1040 is a 3200mAh 3S 11.1V 15C Lithium-Polymer battery pack. The EFLB1040 has a maximum 15C discharge rate where the maximum current draw that we will ever see from the EFLB1040 is 32 amps. Each of the individual cells is rated at 3.7V with a total pack voltage of 11.1V. Thus, the total

amperage capacity of the battery pack would remain at 3200 mAh with a 15C discharge rate.

For the batteries to be charged we are going to build a charging circuit. Our maximum power point tracker will incorporate a Linear Technology LT3652 integrated circuit chip that was solely designed to be a power tracking battery charge for solar powered applications. The LT3652 is a complete monolithic, multi-chemistry battery charger that addresses high voltage applications in a compact design. The key features of the LT3652 are wide input voltage range of 4.95V to 32V with an absolute max voltage of 40V, programmable external charge current of up to 2 amps, resistor programmable float voltage up to 14.4V with 5% charge current accuracy and 0.5% float voltage reference accuracy.

The LT3652 also has an input voltage regulation loop that reduces the current if the input voltage falls below a pre-programmed voltage level set by the voltage divider resistors. The LT3652 also contain provisions for battery temperature. This implemented by using a thermistor during the charging cycle. LT3652 tracks maximum power point to 95% or greater when the charging voltage is set around 12.6V. At higher charging voltages it increase to 98% or greater. The LT3652 is offered in two separate packages from Linear Technology, the DD package, and the MSE package. For our maximum power point tracking charging circuit, we will be using the MSE package. The LT3652 has twelve pins with the right combination of resistors, capacitors, and diodes we can have the LT3652 output the correct voltage and amperage that is needed for our charging circuit as shown in Figure 8.4.4.1.

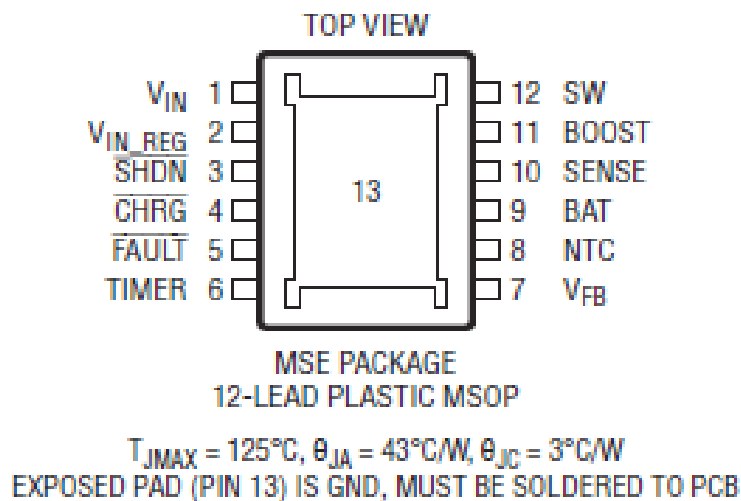


Figure 8.4.4.1 - LT3652 DD Package
“Reprinted with permission of Linear Technology”

Charge current programming is set by choosing an inductor sense resistor. For our particular circuit that we are designing the total expected max current that we would see from the circuit is 462.7mA. The expected value for R_{Sense} would be a resistor with an

approximate value of 0.2161Ω as seen Equation 8.4.4.1 as shown in table 8.4.4.1. Using a resistor divider is needed to program the desired float voltage, $V_{BAT(FLT)}$, for the battery system. In particular, resistors R_{FB1} and R_{FB2} will have to have the correct values to set the 12.6-volt float charge needed in the lithium polymer battery pack. In order to achieve the 12.6V needed we had to use Equation 8.4.4.2 and Equation 8.4.4.3, as shown in table 8.4.4.1, to yield the correct resistors to be used in our design.

The LT3652 also contains a voltage monitor pin that enables it to monitor the minimum amount of voltage coming into the MPPT. The input supply voltage regulation is controlled via the voltage divider resistor R_{IN1} and R_{IN2} . An operating supply voltage can be programming by monitoring the supply through the resistor divider network. This is done by having a ratio of R_{IN1}/R_{IN2} for a desired minimum voltage. In order to achieve the 11.1V needed we had to use Equation 8.4.4.4, as shown in table 8.4.4.1, to yield the correct resistors to be used. The inductor has its own equation to yield the correct value as seen in Equation 8.4.4.5 as shown in table 8.4.4.1. ΔI_{MAX} is the normalized ripple current though the inductor.

The LT3652 also contains a timer termination in which the battery charge cycle is terminated after a preset time. Timer is engaged when a capacitor is connected from the “TIMER” pin to ground otherwise it is directly connected to the ground. A typical timer termination is set to three hours but for our project design we have eliminated this feature from the final design therefore it will be connected directly to ground.

	Principle Equation	Result
Equation 8.4.4.1	$R_{SENSE} = 0.1/I_{CHG(MAX)}$	$R_{SENSE} = 0.2161 \Omega$
Equation 8.4.4.2	$R_{FB1} = (V_{BAT(FLT)} * 2.5 * 10^5)/3.3$	$R_{FB1} = 943 \text{ k}\Omega$
Equation 8.4.4.3	$R_{FB2} = (R1*(2.5*10^5))/(R1*(2.5*10^5))$	$R_{FB2} = 340 \text{ k}\Omega$
Equation 8.4.4.4	$R_{IN1}/R_{IN2} = (V_{IN(MIN)}/2.7) - 1$	$R_{IN1}/R_{IN2} = 12.19 \Omega$
Equation 8.4.4.5	$L = (10 R_{SENSE}/\Delta I_{MAX}) * V_{BAT(FLT)} * [1 - (V_{BAT(FLT)} / V_{IN(MIN)})]$	$L = 38 \mu\text{H}$

Table of Equations 8.4.4.1 Resistors/Inductor Calculations

Below in Figure 8.4.4.5 is the charging circuit that was designed from the setup equations and diagrams. On the left side it shows the input voltage which is the output voltage of the solar panels. The voltage coming in is going through several diodes to prevent any feedbacks. It is also going through several voltage dividers to bring down the voltage to charge the battery which is located at the right bottom corner of the figure. The LT3652 takes the voltage from the solar panels and tracks its maximum current flow to charge the battery with the highest possible current flow. The purpose of the charging circuit is to charge the battery most efficiently through the LT3652.

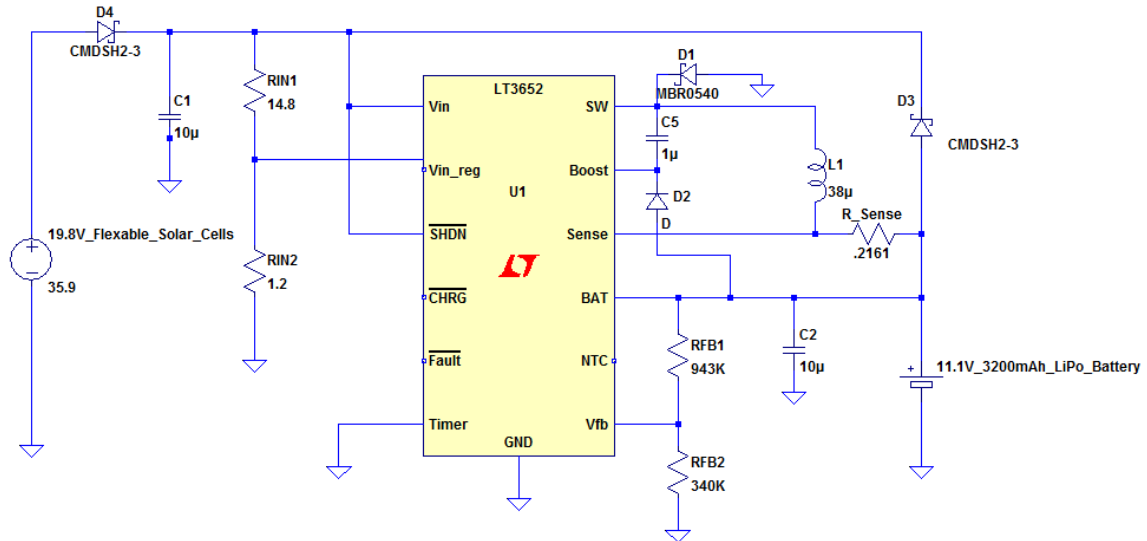


Figure 8.4.4.5 - Charging Circuit

The LT3652, which appears in the figure above, also has status and fault pins that can communicate back a charge or a fault status. The “CHRG” status pin states when the charger is delivering current larger than a C/10 rate or 1/10th of the predetermined charge current. The “FAULT” status pin states when the charger has a bad battery or a NTC fault. Table 8.4.4.2 shows the status pin states that the LT3652 will communicate back through the incorporated diodes.

STATUS PINS STATE		CHARGER STATUS
CHRG	FAULT	
OFF	OFF	Not Charging — Standby or Shutdown Mode
OFF	ON	Bad Battery Fault (Precondition Timeout / EOC Failure)
ON	OFF	Normal Charging at C/10 or Greater
ON	ON	NTC Fault (Pause)

Table 8.4.4.2 – Status Pins State
 “Reprinted with permission of Linear Technology”

As was explained earlier, the design summary of the communication system of the SEV comes in transmitting the collected data by several devices to help keep the SEV watched by the user who is in charge at the ground station. The collective devices are: a dual axis gyro, a single axis gyro, a triple axis accelerometer, a barometric pressure sensor, and a GPS. All of the collective devices are connected to a navigator microcontroller. When the navigator microcontroller collects all the data, it transfers them to the pilot microcontroller. The data then is transferred to the ground station via the RF module which is Xbee Pro 900. The pilot microcontroller is also able to receive orders from the

ground station entered by the user to override the software design of the system. Not only it receives data from Zigbee at 900 MHz frequency range, but also it receives remote control signal at 2.4 GHz frequency range to control the rudder, elevator, and the throttle through the electronic speed controller in case of emergency.

Based on the previous summary, the system is actually built around two microcontrollers: navigator controller and autopilot controller. The navigator controller is connected to the analog devices, which are the dual axis gyro, single axis gyro, and triple axis accelerometer, via its analog to digital converter pins. It is obvious that there are 6 analog outputs which should be connected to the analog to digital connections. Luckily, the ATmega328 supports 6 analog to digital pins which fits the number of analog connections need to install the IDG-500, ISZ-500, and MMA7361L. By design, the navigator controller is also connected the barometric pressure sensor, GPS, and the autopilot controller. The barometric pressure sensor is interfaced with I²C protocol, two-wire interface, which makes it easy to be connected. Therefore, the BMP085 is connected via the two serial interface pins (serial data line and serial clock line.) The GPS, on the other hand, is interfaced with a universal asynchronous receiver and transmitter serial ports. Therefore, the Copernicus GPS receiver is connected via the UART to the ATmega328. Finally, the navigator controller is connected with the autopilot controller via the two wire interface to transmit all the collected data to the master ATmega328 microcontroller. The autopilot controller is also connected to several components which are the remote control receiver, Zigbee 900 transceiver, two servos, and brushless motor via electronic speed controller. The RC receiver is connected to control the rudder and elevator through the servos; it also controls the throttle on the brushless motor. The RF module is interfaced with UART serial pins; thus, the Xbee Pro 900 is going to be connected similar to the GPS connection with the navigation controller. Hint, the RC receiver and the RF transceiver operate at different frequencies to make sure they do not overlap. The autopilot controller is also outputting signals to control the servos and the brushless motor. Figure 11.2 shows the communication system of the SEV as described in the design summary.

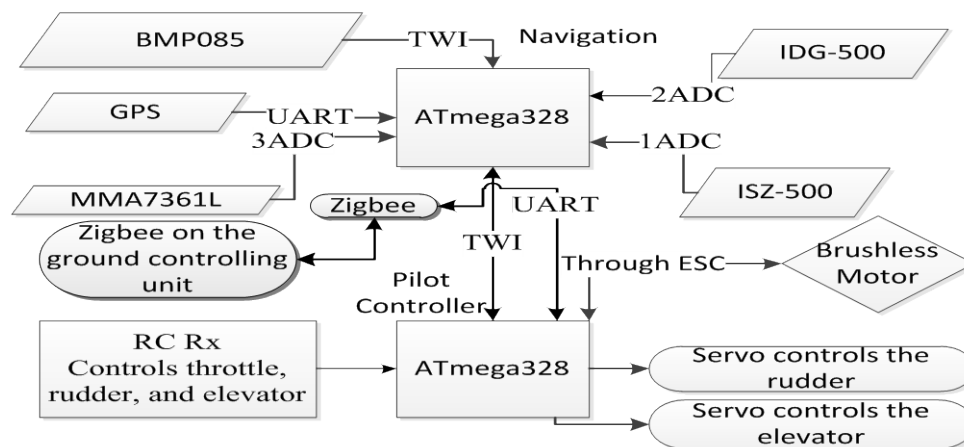


Figure 11.2 - the Communication System Design Summary

For the hardware to run as we would like, we will need to utilize several different types of software for this project. We will need a software to run fly the UAV, this software will also need to monitor the battery. We will need software to navigate the UAV, plus we will be using a program to set the waypoints for the UAV. After this we will also use a simulator for testing in the future. For this project, we will be using software called ArduPilot version 2.7. This is an open source autopilot platform created by Chris Anderson and Jordi Munoz of DIY Drones. The code that will be used for the flight of the S.E.V. was developed by Jason Short and Doug Weibel. There is also a desktop utility that will be used for this project. The desktop utility is used for entering the waypoints and the home location for the S.E.V. Both the autopilot and desktop utility are published under an LGPL license that allows free use and modification as long the resulting product is open source and retains the DIY Drones attribution is retained. The ArduPilot allows for programmable waypoints using the desktop utility. These are the destination that the S.E.V. will fly to until it returns to the home location; that is also entered through the desktop utility. The waypoint and home location are usually set up before the flight takes place where an internet connects can be obtained. The internet connection is use to retrieve, when select in the utility, the altitude at the waypoints selected. This option can also be turned off and an altitude can be entered for the S.E.V. to try and maintain.

This function will not be used for this project because we are going to alter the code to have the S.E.V., during its flight between waypoints, glide from a hardcoded maximum altitude to minimum hardcoded altitude. Then increase the throttle to bring the S.E.V. back to the maximum altitude, then cut the throttle and repeat the cycle over. The waypoints and home location are stored on the EEPROM along with the sensor calibration. This is done in case of a system restart. ArduPilot will control the elevator, rudder and throttle. It also supports “fly-by-wire”, which allows the user to tell the S.E.V. where to go and the ArduPilot will fly the plane; this is very important when inexperienced flyers take control. Under the ArduPilot_2_7 tab is where you will find the initialization of the program, there is a main loop that initializes three very important functions, of different speeds, for this program; these functions are loops themselves and are labeled as such: main loop event, medium loop, and fast loop. Below in Figure 9.2.1 is a graphical visualization of the flow of the main loop. The main loop is designed to run as fast as possible and we are looking to reach about 50Hz

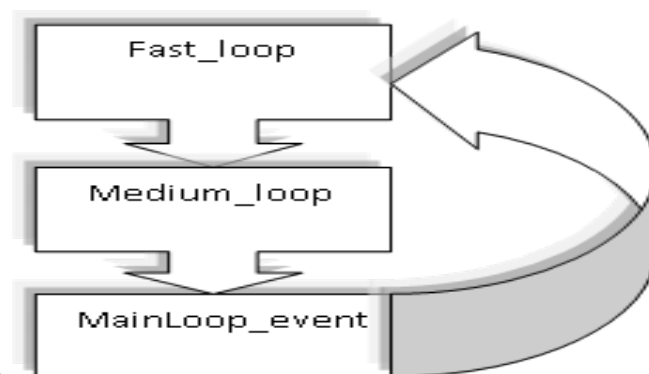


Figure 9.2.1 - Main Loop

In the main loop the last function that would have been called after each case of the medium loop was finished running. Below in Figure 9.2.4 is a simple layout of the altitude gliding code. This code may change before the end of the project.

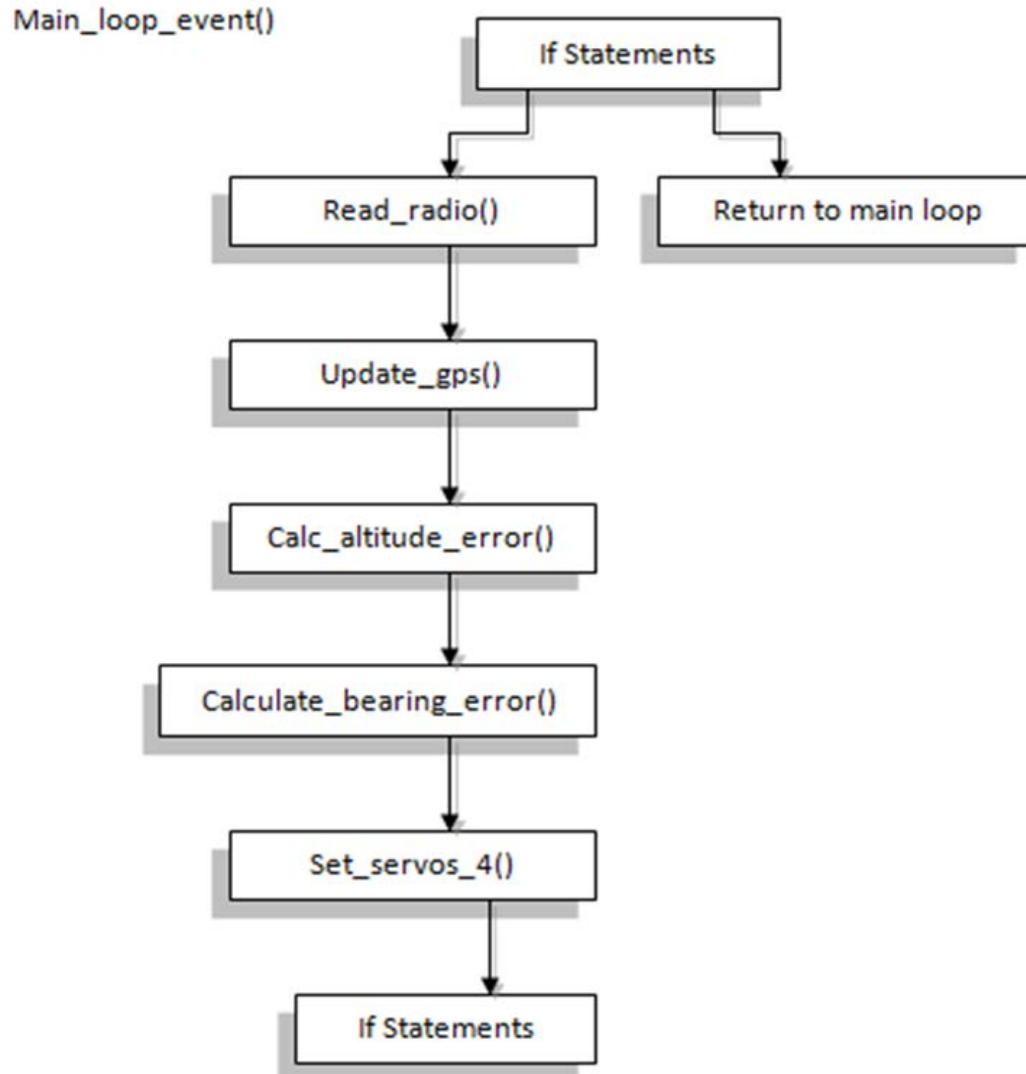


Figure 9.2.4 -The main loop event for allowing the UAV to glide.

First we will be using an ‘If’ statement to check the current altitude against our set minimum altitude. If the altitude is greater than minimum altitude value then the function will end and return to the main loop. But if the altitude was less than the set minimum value, then it would call on the functions required for returning the UAV back to its maximum altitude. First, before returning the UAV to the maximum altitude, it will run the read radio function. It does this to avoid overriding the user, if the user is manually controlling the aircraft. After checking to make sure that the user is not in control; the next function called is the update GPS. This is called to refresh the altitude and location data, so that we are not using old data to adjust the UAV’s altitude. After the update, we

run an altitude error check, this is so that we can adjust the servos accordantly; doing this will result in the UAV correcting its altitude. Then we will need to check if our bearing is correct; so we will then need to call calculate bearing error function. This function will check, as it name implies, if the UAV is off course and the then correct its bearing. The next function to be called is set servos function. This function will write out the servo PWM values. Last in our code is a set of 'if' statements to check if we are using a simulator or are using a GPS. This is because the program needs to know where to write out the values, into a simulator or to the ground control station.

The ArduPilot will relay on the Altitude Heading Reference System code, this will be loaded onto another microcontroller. The AHRS code is developed using the Arduino IDE; the code will be used to maintain a model of the UAV's orientation in space. This code is based on Bill Premerlani's Direction Cosine Matrix (DCM) algorithm . The DCM is a 3 by 3 matrix array; the gyro data is used in a time step integration to update the matrix. With this data from the sensors and GPS can be used to correct for errors. These errors are caused by numerical errors. With this data we can determine pitch, roll and yaw; but for this project we will only be using the pitch and yaw. With this code we are now able to use the above coding to fly, without the control of a user, our UVA to waypoints. Plus, with code modification that we have designed, we will be able to extend the flight time of a RC aircraft.

The ArduPilot mission Planner is a configuration tool to help you easily plan your UAV's missions. Missions are a set of waypoints for the UVA's flight path. Below in Figure 9.4.2 is the interface of the mission planner. You will notice the mission planner uses Google maps for laying down the waypoints; on the map you can see a home point. This home can be clicked and dragged to any location you require. The blue balloons are the waypoints that the UAV will fly to.

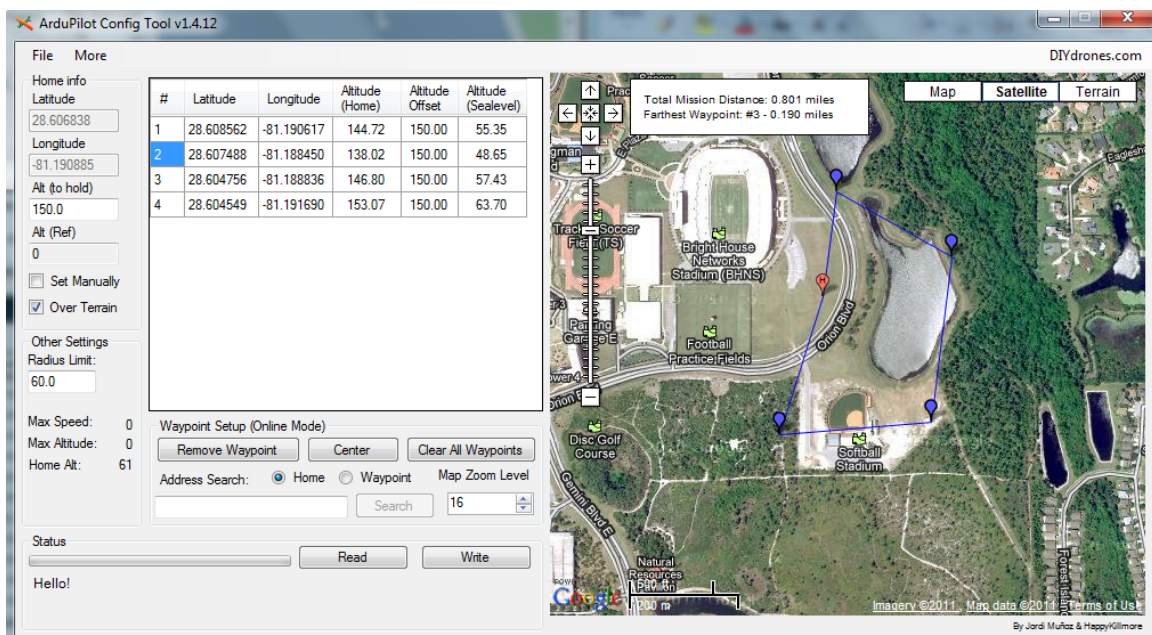


Figure 9.4.2 - Mission Planner with waypoints selected

During the flight of our UAV we will be monitoring it using the ground control station. The ground control station is an addition to the project that uses ZigBee modules to communicate to the UAV. The Ground control station is a open source software developed by DIYdrones that utilizes Google Earth. During the flight of the UAV the ArduPilot code creates a log of the location and altitude; and that is sent to the ground station using the Xbee modules, where it is updated on Google Earth.

11.1 Block Diagram

The following block diagram serves as an organized summary of our design as a whole. This diagram combines all of the components from each section to show their correlation with one another. These sections can be broken down into four distinct categories: The airframe, the autopilot navigation system, the communication network, and the battery charging component.

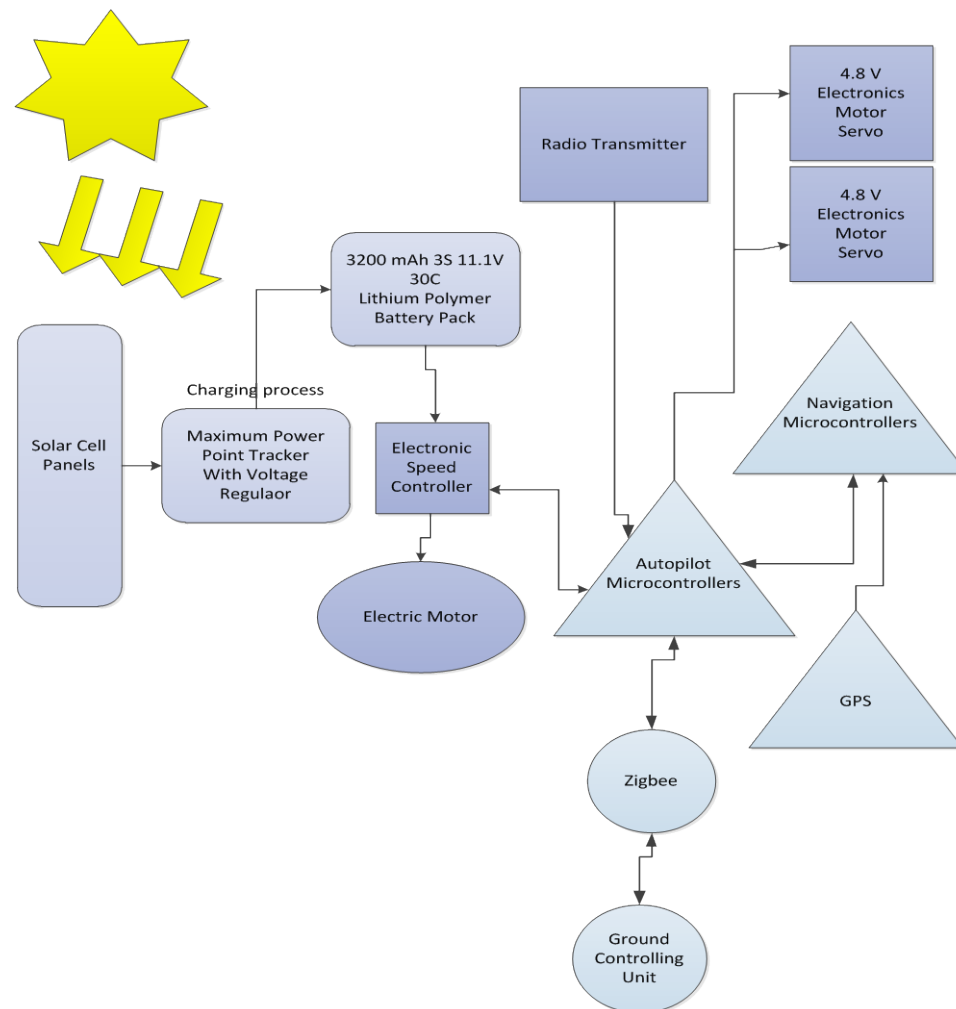


Figure 11.3 - Block Diagram

Chapter 12: Prototyping

As we approach our final design, we will begin construction on our project. However, we will not begin with the immediate construction of our intended final design. There is a great deal of testing that will be performed prior to this. The construction process will be divided into three phases of testing. The first phase involves building a radio-controlled car which will be used to test the majority of the electronic components of our design before we attempt a test flight. The second phase will be our first flight test with a smaller version of the final plane. The third phase of testing will combine the components of the previous tests and enable us to begin construction on our final design.

12.1 Construction

After the first two phases of testing, we will commence construction on the final S.E.V. design. The airframe can be constructed by using the detailed instructions from the ProJet RQ-11 manual. This instruction manual serves as a detailed set of steps to describe the assembly of the airframe. However, it does not explain the connections of the electrical components. This process will be determined by our group.

The fuselage of the RQ-11 is quite spacious and will contain the majority of the added payload. A total of two circuit boards will be placed inside the fuselage. One of the circuit boards is the charging circuit and the other contains various electronic components such as the microcontrollers, GPS, gyros, accelerometer, and pressure sensor. The fuselage is shown below in Figure 4.6.1.



Figure 12.1.1- Inside the fuselage of the UAV

As stated earlier, we will have two boards total. One for the charging circuit and the other will contain the necessary components for the autopilot system. The boards will be layered with one over the other to form a stacked appearance. These two will be stack above the battery. In Figure 12.1.2, we can see the two PCBs being stacked To do this we will build a mount that will rest on top of the battery, while letting the battery hold the mount secure. Below in Figure 12.1.2 is a rough sketch of that mounting system.

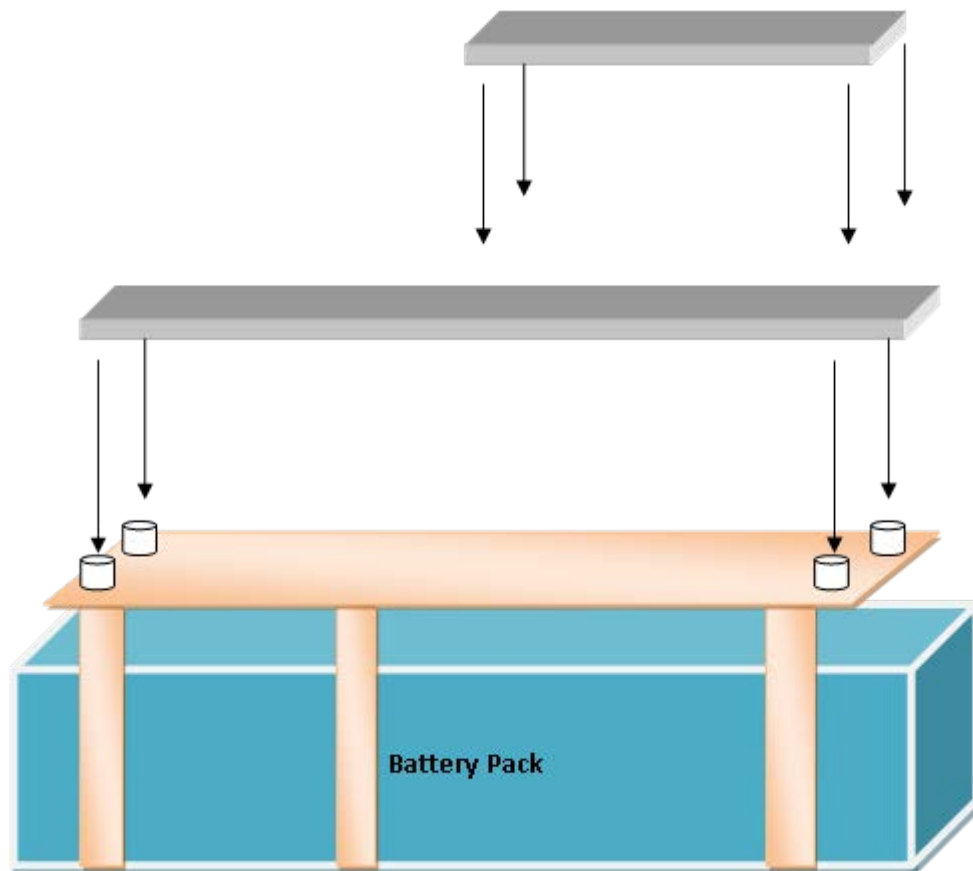


Figure 12.1.2 - The battery back pack mounting of the PCBs

In Figure 12.1.2, the two dark gray pieces are the PCBs; these are connected through the gold pin connectors that will be placed on them. These connectors are not represented in this diagram. The two PCBs will then rest on the four white spacers to keep the board of the mount. The mount will be constructed with balsa wood and wood glue. Balsa wood is very light weight, thus making it a good material to use during this project. The mount will not rest directly against the battery; it will be raised up allowing the battery to cool during operation. This mount will be connected to the battery by Velcro. Then attached to the bottom of the battery are two strips of Velcro, which will attach the whole mounting system to the fuselage. The battery's weight plus the Velcro will keep the PCBs stable and level; which is important when calibrating the sensors.

We begin construction on the RQ-11 by putting together all of the assorted components. For the most part, construction of the RQ-11 is quite simple. Many of the connections simply require glue to be held together. To assist in the set up process, the ProJet RQ-11 kit comes with a detailed manual that provides guidelines for putting the parts together. The basic parts for the airframe are shown below in Figure 12.1.3. The parts include: the three sections of the main wing, the fuselage, the rudder fin, and the elevator fin.



Figure 12.1.3 - Assorted parts of the RQ-11

The following steps explain the different steps in construction as well as what electronic components will be connected in each section of the plane. They provide summarizations of construction based on the sections of the plane. Additionally, they will explain how each electronic component connects together.

One important aspect of the construction process will involve the placement of the solar cells. The cells will be fitted on top of the main wing. Our plan is to have a total of 10 solar cells on the aircraft. Eight of them will be arranged in a 2 x 4 array while the last two will be placed vertically to maximize the use of the allotted space. The solar cells will be connected to the battery for the charging system.

Pictured below is Figure 12.1.4 to show the main wing of the RQ-11. The solar cells will be layered across the wing in the manner that was described previously. The wing was shipped in three different sections that will be held together by two dowel rods and two 2in metal rods. One dowel rod and one metal rod on each side of the middle piece of the wing will hold the other two sections together.



Figure 12.1.4 - Main wing of the RQ-11

Figure 12.1.5 shows the whole airframe and can be used as a reference to where all of the components will be attached during construction. The wing will be attached to the top of the fuselage by two screws. These screws allow the wing to be easily removed; this makes the plane much easier to transport.

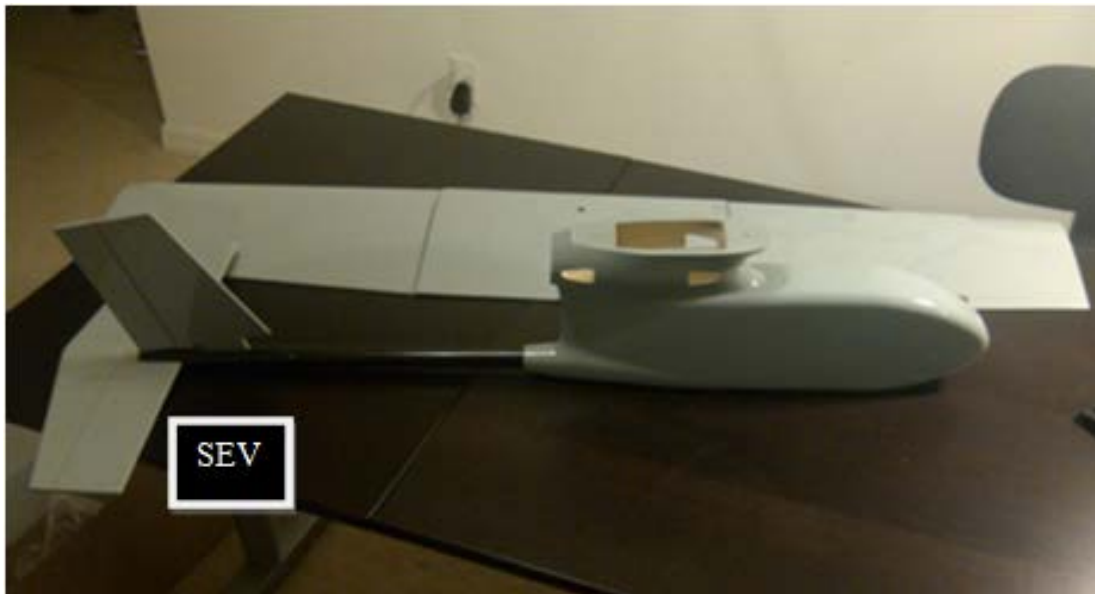


Figure 12.1.5 - Partially built RQ-11 airframe

The section above the fuselage is the area of the aircraft that will contain the motor. This area will be referred to as the prop tower. The propeller will be attached to the motor and will be exposed just outside the prop tower. The main wing will then be placed above the prop tower to cover the motor. The image below shows the top view of the prop tower which will provides access to the motor.



Figure 12.1.6 - Top view of the prop tower

Shown below in Figure 12.1.7 is the bottom view of the fuselage under the prop tower. This area of the plane will contain the two servos that control the rear rudder and elevator.

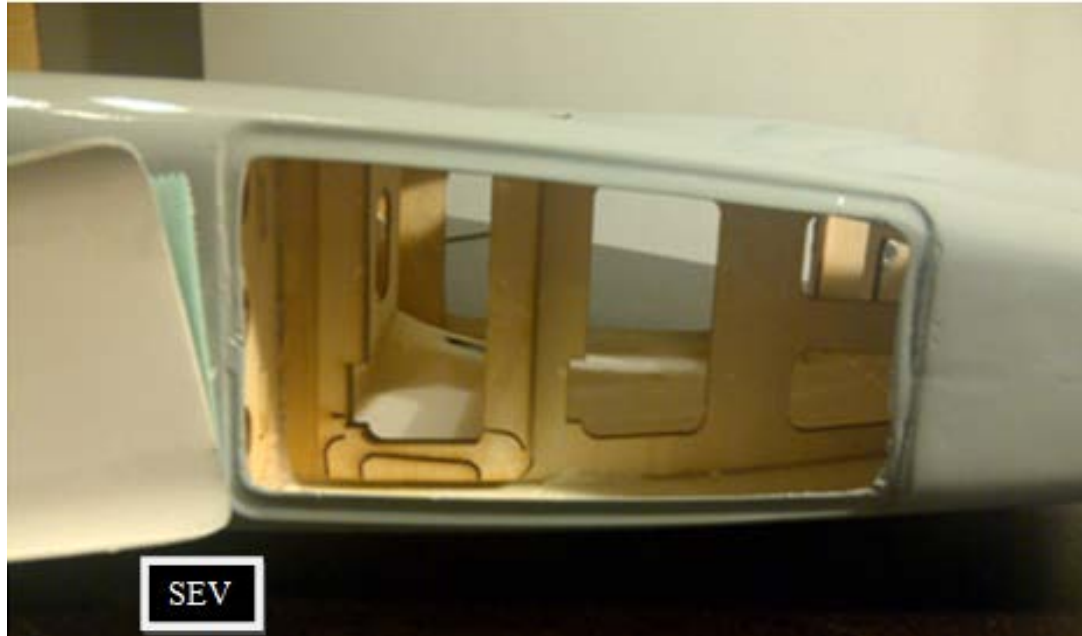


Figure 12.1.7 - Bottom view of the fuselage

The servos can be accessed through the underside of the airframe. The design will include a total of two servos to control the rudder and elevator located on the tail of the plane. The last section of the airframe is the tail. The tail of the plane contains two

significant components: the rudder and the elevator. The rudder runs vertically and is attached to the tailfin while the rudder lies horizontally. Together, these two components directly affect the maneuverability of the plane during flight. A picture of the tailfin can be seen in the previously displayed image of Figure 12.1.6.

The cylinder that connects the tail to the fuselage will be used for connection of the servos and the rudder/elevator. We will use cables that connect one servo to the rudder and another to connect the other servo to the elevator. This interconnection is what allows the servos to move the rudder and elevator during flight. This type of cable connection is known as a pull-pull system. This system uses a total of four cables with two cables attached to each servo. For instance, the connection between the elevator and the servo will have one cable running to one side of the elevator and another cable is attached to the other side. The tensions for the two cables are equal, so a pull on one end is equal to the same pull on the other side. Using the same method, the other servo is connected to the rudder.

Shown below is the tailfin of the RQ-11. From Figure 12.1.8 it can be observed that the rudder and elevator are already connected to their respective fins.



Figure 12.1.8 – Tailfin of the RQ-11

Chapter 13: Project Prototype Testing

13.1 Introduction

Before we can just be flying there are a few things that need to be tested so that we do not damage or destroy any of our vehicles and electronics. This section refers to the process of testing our design by using real-world applications that can provide measurable data to prepare the finalized version of our project. The testing process will include two prototypes that will be built before the final design. The first prototype will be a small RC plane with no solar cells attached. The second prototype model will be our final plane with the solar cells connected. The plane will test the GPS component and provide feedback on how well the autonomous flight system performs. Below is a list of the different steps that will be taken in the testing phase.

- Initial code configuration and simulation
- Test of modified code through simulator
- Test of solar cells and charging circuit
- Test of project PCBs through simulator
- Test of solar cell and project PCBs to verify charging of battery
- First flight test: small airframe, no solar cells
- Second flight test: final airframe, no solar cells
- Third flight test: final airframe with adjustments, no solar cells
- Fourth flight test: final airframe with adjustments, with solar cells
- Final flight test: final airframe with adjustments and with solar cells

13.2 Testing

The first sets of tests we will be doing are software test. Below in Figure 13.3.1 is a picture of the development boards that we will use to test the code. The first thing in testing the code is getting the software configured to the type of airframe we are using. What is meant by this, is that the program needs to know the tail configuration and if we are using ailerons or not. The green board in the photograph will have the navigation code loaded onto it. The red board will be running the auto pilot code. Once we have the code configured correctly we will load it to these boards and confirm that the code operates as it should. To check this we will be using a simulator. The simulator will provide the GPS data to the board and then the board will tell the plane how to fly. After we have the code configured and know that it is working as it should. Then we will test the modified code that we are inserting into the ArduPilot code. This also will be tested through the simulator.



Figure 13.2.1 - Dev boards purchased from Sparkfun.com

While the testing of the code is underway, the other team members will be testing the solar cells. For this, they will be testing and confirming the total output of the arrays, and they will also be check for the best array configuration. Plus, they will be testing that the charging circuit is reaching the correct output. When the PCBs that were designed have order and received, we will then test these boards. First we will do a visual inspection of all the parts solder to the board; whether they have been solder by us or a vendor. After confirming the build quality, then the software will be loaded to the board. The board then will be tested using simulator software to verify that the code is working correctly with the new autopilot PCB. Meanwhile, the power PCB will be connected to the solar cells to verify that the battery is charging from the cells. Then the two PCBs will be connected together to confirm that the power PCB will power the autopilot board. The two boards will be tested again using the computer simulator. Once we believe that the two boards are working correctly, we will then install them into the UAV for flight testing.

During the next phase of prototype testing we will be using a smaller airframe with similar features as our Raven RQ-11 plane. We are doing this because if there is an incident where the plane was to lose control and crash; it would have happened to a much cheaper airframe. This plane will provide feedback on the navigation and autonomous flight capabilities. It will also provide additional input on the GPS features. By utilizing the smaller test plane, valuable information can be gathered that will provide input for any minor adjustments that need to be made before building the final S.E.V. design. This testing will be crucial for determining the accuracy of the modified code. Also, it will allow us to observe how the plane flies without human control. A significant amount of information can be gained just from observation alone. This will give us input on how well the rudder and elevator responds to the guided control from the GPS directions.

Once we have made the some adjustments and are comfortable with how the plane handles; we will install the system into our final airframe, the RQ-11. Then we will do a

test flight of that setup. This test flight will not include the use of the solar cells. This is because, again, if we have an incident and crash the plane; we will not have to replace the cells. The purpose of this test will be to fine tune the airframe and servos so it will fly smoothly. These earlier test will also provide us with how long the battery will last under normal conditions. After this test and we iron out the small details, we will then mount the solar cells to the wing and perform another test flight. It should be noted that all these test flights will be don at a low altitude to help prevent unnecessary damage to the UAV. The last test flight will provide us with the difference in flight time of the UAV with solar cells. Here we will make adjustments to help the flight time.

13.3 Rules & Regulations

Due to the nature of our project, it is important to make sure that all federal rules and regulations are followed. More specifically, it is important to be aware of the Federal Aviation Administration (FAA) regulations that pertain to the flying of personal radio-controlled aircraft or general Unmanned Aerial Systems (UAS). The FAA defines an unmanned aircraft as any aerial vehicle that does not have an on-board pilot. This description provides a wide range of vehicles with wingspans ranging from less than 1 foot to well over 200 feet. Since, our design falls into this broad category, it is imperative that we take caution in exercising the FAA rules.

For the most part, the Federal Aviation Administration's regulations are not strict and are mostly comprised of common-sense type guidelines. For example, the FAA encourages individuals to avoid crowded areas and busy streets while controlling UAVs. The more detailed rules of the FAA depend on whether the UAV is categorized as a civil aircraft or a public aircraft. The development of our S.E.V. design falls into the model aircraft category since it is a self-made project and is intended only for personal use by our group. The regulations laid out by the FAA encourages good judgment on behalf of the user so as not to endanger the well being of other spectators that may be in the area. Additional guidelines for appropriate site selection include avoiding areas that are sensitive to noise such as churches, schools or residential areas. One of the significant guidelines is to remain below 400 feet from the ground level to avoid other aircraft in flight. Furthermore, the user is expected to maintain a line-of-sight awareness on the vehicle for the entire duration of the flight.

13.4 Test Facilities

Finding an appropriate location that can allow for prototyping flight testing is a necessary part of our project. In order to avoid problems with obstacles, an open grass field is the best suitable location for flying our UAV. It should be taken into consideration that any flight testing location should be clear of vehicular traffic, power lines, and crowds. Another important factor to consider is the wind conditions. Small aircraft are highly susceptible to wind and should not be flown in such conditions. There are numerous areas around the UCF campus that meet the above criteria.

There are numerous areas around campus with enough open space to test out our prototypes. Currently, there are not any strict rules or regulations regarding flying small UAVs in public areas. The only guidelines are against flying in areas where there is a high chance of damaging private property. However, public parks are ideal locations for flight testing.

13.5 Flight Path

During the second phase of testing with the smaller plane, we will use a designated flight path for the GPS guided navigation to follow. Shown below is a Google Earth screenshot of the UCF campus which shows a possible test flight route. The image shows the starting point just below the UCF arena. The path continues in a clockwise circle and ends near the starting location. This picture below shows one of the possible test flight paths that will be simulated before flight. This also shows the intended range of our plane for a typical flight.



Figure 13.5.1 – Flight Path

Chapter 14: Administrative Content

14.1 Budget and Financing

One of the initial steps of our project involved organizing a summarized budget that we could compile together to create an approximate cost total. As we continued to research components for our design, we were able to closely adjust the specific costs to more accurately reflect the actual final cost. The table below shows the detailed budget estimate, arranged by category.

Description	Quantity	Price Each	Total price
<i>Airframe</i>			
Plane	1	\$ 250.00	\$ 250.00
Motor	1	\$ 70.00	\$ 70.00
Servo	2	\$ 50.00	\$ 100.00
Electronic Speed Controller	1	\$ 60.00	\$ 60.00
<i>Electronics</i>			
Microprocessor	1	\$ 4.25	\$ 4.25
Triple-Axis Accelerometer - MMA7361L	1	\$ 7.95	\$ 7.95
Barometric Pressure Sensor - BMP085	1	\$ 8.95	\$ 8.95
Dual Axis Gyro - IDG500	1	\$ 24.95	\$ 24.95
			\$ -
<i>Communication</i>			
GPS	1	\$ 55.00	\$ 55.00
Transmitter	1	\$ 30.00	\$ 30.00
Receiver	1	\$ 30.00	\$ 30.00
ZigBee	2	\$ 50.00	\$ 100.00
<i>Power</i>			
Solar Cells	10	\$ 27.45	\$ 274.50
Lithium-ion Polymer Battery Pack	1	\$ 60.00	\$ 60.00
Charging Circuit	1	\$ 50.00	\$ 50.00
Repair Parts			
	1	\$ 100.00	\$ 100.00
Sub-Total			
			\$ 1,225.60
Tax		6.5%	\$ 79.66
Total			\$ 1,305.26

Table 14.1.1 Budget and Financing

14.2 Milestones

Over the course of our project organization, we were able to lay out a general overview of designated milestones. These milestones were intended to provide a straightforward checklist that could be used to measure our progress over the duration of the project development. Listed below are the milestones corresponding to both Senior Design 1 and Senior Design 2, respectively:

- SD1* - Preliminary testing of the programming for the microprocessor
- End of SD1 - Acquisitions of electronic parts
- End of SD1 - Preliminary conceptual design of the electronics board
- First week of SD2* - Testing of the circuit on Multisim or LTSpice
- Second week of SD2 - Wiring schematic
- SD2 - Electronic prototype board
- Fourth week of SD2 - Hardware prototype
- SD2 - Final hardware fabrication
- SD2 - System integration onto the final test subject
- SD2 - Controlled flight test with all system integrated (Weight Test)
- SD2 - Final flight test
- SD1-in the time frame of senior design 1
- SD2- in the time frame of senior design 2

14.3 Milestone Chart

The purpose of a milestone chart is to provide an arranged schedule in the form of a chart to display the information from the milestone list. This allows our group to more easily distinguish the predetermined goals for the project. This chart in particular, contains checkpoints over the duration of July and August through which we carried out the corresponding tasks. The described milestone chart is displayed below:
















































		Task Mode	Task Name	Duration	Start	Finish
0			Senior Design I Project Timeline	35 days?	Mon 6/20/11	Fri 8/5/11
1			Airframe Research	31 days	Mon 6/20/11	Sun 7/31/11
2			Airframe Research	12 days	Tue 6/21/11	Wed 6/6/11
3			Research Electric Motor	10 days	Thu 7/7/11	Wed 7/20/11
4			Research Electric Servos	8 days	Thu 7/21/11	Sun 7/31/11
5						
6			Communications	31 days	Mon 6/20/11	Sun 7/31/11
7			Research Zigbee	7 days	Tue 6/21/11	Wed 6/29/11
8			Research Accelerometer	7 days	Wed 6/29/11	Thu 7/7/11
9			Research Gyroscope	7 days	Wed 7/6/11	Thu 7/14/11
10			Research Barometric Pressure Sensor	7 days	Wed 7/13/11	Thu 7/21/11
11			Research GPS	7 days	Fri 7/22/11	Sun 7/31/11
12						
13			Power Systems	31 days	Mon 6/20/11	Mon 8/1/11
14			Research Solar Cells	11 days	Mon 6/20/11	Mon 7/4/11
15			Research Battery	9 days	Tue 7/5/11	Fri 7/15/11
16			Research MPPT	12 days	Sat 7/16/11	Sun 7/31/11
17						
18			Software Design	31 days	Mon 6/20/11	Sun 7/31/11
19			Research ArduPilot	10 days	Tue 6/21/11	Mon 7/4/11
20			Research GCS	10 days	Tue 7/5/11	Mon 7/18/11
21			Research Flight Simulator	5 days	Mon 7/18/11	Fri 7/22/11
22			Research IMU	8 days	Thu 7/21/11	Sun 7/31/11
23						
24			Senior Design Paper Formatting	2 days	Sun 7/31/11	Mon 8/1/11
25			Senior Design Paper Binding	2 days	Wed 8/3/11	Thu 8/4/11
26			Senior Design I Paper Due	1 day	Fri 8/5/11	Fri 8/5/11

Figure 14.3.1 - Summer 2011 Gantt Chart

This next chart is a continuation of Figure 14.3.1 and shows our projected milestone for the Fall semester schedule. In contrast to the previous milestone, the goals for the Fall consist of the testing and construction of our design as opposed to researching. The chart below shows the milestones for Senior Design II.






































ID		Task Mode	Task Name	Duration	Start	Finish
1			Senior Design II Project Timeline	90 days	Mon 8/8/11	Fri 12/9/11
2			Acquire Airframe Components	20 days	Mon 8/8/11	Fri 9/2/11
3			Acquire Communication Components	20 days	Mon 8/8/11	Fri 9/2/11
4			Acquire Power System Components	20 days	Mon 8/8/11	Fri 9/2/11
5			Acquire Software Components	20 days	Mon 8/8/11	Fri 9/2/11
6			Communication System Assembly	14 days	Fri 9/2/11	Wed 9/21/11
7			Communication Pre-Test	3 days	Thu 9/22/11	Mon 9/26/11
8			Solar Panel Pre-Test	2 days	Fri 9/2/11	Mon 9/5/11
9			Battery Pre-Test	2 days	Tue 9/6/11	Wed 9/7/11
10			MPPT Charge Controller Build	20 days	Thu 9/8/11	Wed 10/5/11
11			MPPT Charge Controller Pre-Test	3 days	Thu 10/6/11	Mon 10/10/11
12			Software Programming	30 days	Fri 9/2/11	Thu 10/13/11
13			Software Simulation	5 days	Fri 10/14/11	Thu 10/20/11
14			Software Testing	3 days	Fri 10/21/11	Tue 10/25/11
15			PCB Design	9 days	Wed 10/26/11	Mon 11/7/11
16			Final Software Testing	4 days	Tue 11/8/11	Fri 11/11/11
17			Airframe Assembly	33 days	Fri 9/2/11	Tue 10/18/11
18			Airframe Pre-Flight Test	10 days	Wed 10/19/11	Tue 11/1/11
19			Airframe Initial Test Flight W/O Components	7 days	Wed 11/2/11	Thu 11/10/11
20			Airframe Systems Integration	1 day	Mon 11/14/11	Mon 11/14/11
21			Solar Panel Integration	1 day	Tue 11/15/11	Tue 11/15/11
22			Battery System Integration	1 day	Wed 11/16/11	Wed 11/16/11
23			MPPT Charge Controller Integration	1 day	Thu 11/17/11	Thu 11/17/11
24			Communication System Integration	1 day	Fri 11/18/11	Fri 11/18/11
25			Power System Ground Test	1 day?	Mon 11/21/11	Mon 11/21/11
26			Software Ground Testing	1 day?	Tue 11/22/11	Tue 11/22/11
27			Communication Ground Testing	1 day?	Wed 11/23/11	Wed 11/23/11
28			Pre-Flight With All Systems Integrated	4 days	Thu 11/24/11	Tue 11/29/11
29			Final Flight With All Systems Integrated	4 days	Wed 11/30/11	Mon 12/5/11
30			Final Flight Presentation	4 days	Tue 12/6/11	Fri 12/9/11
31			Senior Design II Paper Formatting	2 days	Sat 11/26/11	Sun 11/27/11
32			Senior Design II Paper Binding	2 days	Mon 11/28/11	Tue 11/29/11
33			Senior Design II Final Paper Paper Due	2 days	Thu 12/1/11	Fri 12/2/11
34			Senior Design II Presentation	2 days	Thu 12/8/11	Fri 12/9/11

Figure 14.3.2 - Fall 2011 Gantt Chart

Chapter 15: Senior Design II Modifications

15.1 Motor Mount

Our overall airframe structure for the S.E.V. was based on the Projet RQ-11. Everything that was included with the RQ-11 frame kit was used as instructed by the manual with one exception. During the testing phase, we discovered that the motor mount for the Rq-11 was unstable and it lacked the strength to effectively hold our motor. This was realized after our original motor dismounted itself from the plane while it was powered on. In response to this incident, we reinforced the motor mount with 4 additional layers of plywood which were attached to a metal mount. Each layer was held together with epoxy glue and screws. It is highly recommended to anyone who uses the Projet RQ-11 in the future to reinforce the motor mount in a similar fashion. A picture of the motor mounted to our plane is shown below.



Figure 15.1.1 Motor mount before and after enforcement

15.2 Motor

The E-Flite Power-10 motor was a very compatible selection for our RQ-11 airframe. This is an out-runner type motor, which is quite efficient for our 10 x 5 propeller size. This motor is rated at 1100Kv with a continuous current rating of 32A and a burst current of 42A. We had originally chosen an Electrify Ammo 28-35-5100 in-runner motor. We now know that in-runner motors with high Kv ratings such as 5100Kv are best used for planes with small propellers. This motor spun too fast with our 10 x 5 propeller and disconnected itself from the motor mount, causing slight damage to our plane. Immediately after this incident, we reinforced the motor mount and replaced the motor with the Power-10 out-runner. We also learned that lower Kv ratings generally produce higher torque. This means that while the Power-10 may spin slower, it actually produces more power than the Ammo 28-35-5100. For our application, the higher torque meant that the Power-10 would accelerate faster than the Ammo 28-35-5100. The Specifications for the E-Flite Power-10 1100 brushless out-runner motor are listed in the table below.

Model #	E-Flite Power-10
Recommended Prop Range	10x5 ~ 12x6
RPM/Volt	1100 Kv
Max Voltage	7.2-12V
Resistance	0.043 ohms
Idle Current	2.10A @ 10V
Continuous Current	32A
Max Burst Current	42A
Cells Li-Po	2-3S
Weight	122g (4.3 oz)
Overall Diameter	35mm (1.40 in)
Shaft Diameter	5mm (0.20 in)
Overall Length	43mm (1.60 in)
Type	Brushless outrunner motor

Table 15.2.1 Power-10 Specifications

Figure 15.2 shows the motor that we are currently mounting on our enforced motor mount. This approach enables us to mount an out-runner motor.



15.2.1 Power-10 Motor

15.3 Electronic Speed Controller

The S.E.V. Project uses the E-Flite 60A Pro SB Brushless ESC. This speed controller has a continuous current rating of 60A and a max burst current rating of 75A. This is easily compatible with our 30A motor. We chose this ESC because the high current rating would allow us to switch to a more powerful motor if it was deemed necessary. Additionally, the weight of this electronic speed controller was not much higher than one with a lower current. Furthermore, the E-Flite brand utilizes a unique EC3 battery connector which allows for convenient connecting and removing of the battery. A table of specifications is shown below for the E-Flite 60A Pro SB Brushless ESC.

Model #	E-Flite 60-Amp Pro SB Brushless
Max Continuous Current	60 A
Max Burst Current	75 A
BEC Output	5V/2.5A
Input Voltage	3-6 cells (LiPo)
Weight	66g
Dimensions	33x76x13mm

Table 15.3.1 E-Flite 60-A Pro SB Brushless Specifications

The Figure displayed below shows the ESC that we are currently using on our SEV.



Figure 15.3.1 The E-Flite 60A

15.4 Circuit Boards

We have two circuit boards a printed circuit board and a point to point circuit board. We

went by doing this to be experienced in both designing PCB on EAGLE software and soldering PTP circuit boards.

15.4.1 Autopilot Board

As presented in the figure below, this board was designed on EAGLE. Its width is 5cm and its length is 12.5cm. The Autopilot board design is composed of two layers, a ground layer which acts as a ground bus and a positive layer which is the upper side of the board. The board designed to be symmetric as if it is two separate boards brought together. The left side of the board is the navigation unit. The navigation unit is composed of the Atmega328, single and dual axis gyroscopes as well as a triple axis accelerometer. It also has a connection to the GPS. The right half side of the board is the autopilot unit. The autopilot unit composed of two microcontrollers, Atmega328 and Attiny45. It also has a multiplexer as well as connections to the electronic speed controller, servos, and the Xbee transceiver.

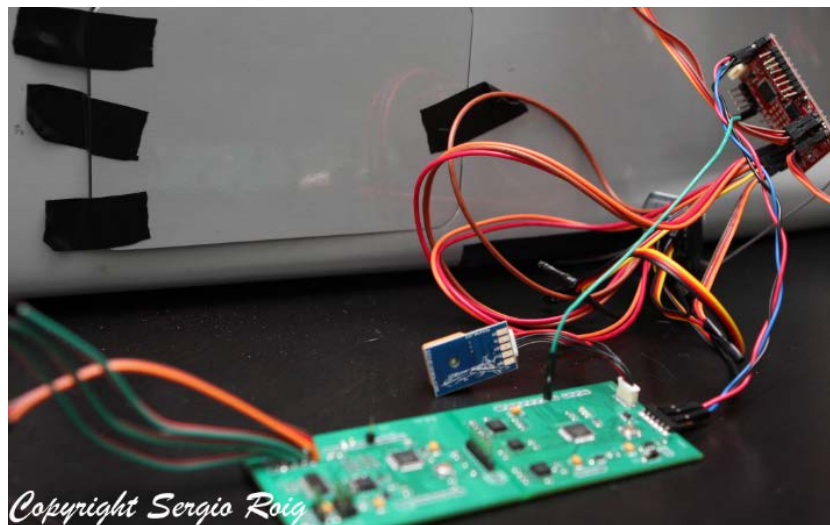


Figure 15.4.1 The Autopilot Board

15.4.2 Charging-Circuit Board

The charging circuit board was designed to be a point to point circuit board. Its width is 5.2cm, and its length is 11.5cm. The charging circuit has the LT3652 which is the maximum-power point tracker. For the PTP to work with the LT3652 we had to have the LT3652 mounted on a separate board with gold pin connectors so it would fit the through holes on the PTP board. The rest of the components were through hole as well. This board has connections to the battery and the solar panels. The figure below shows the charging circuit board.

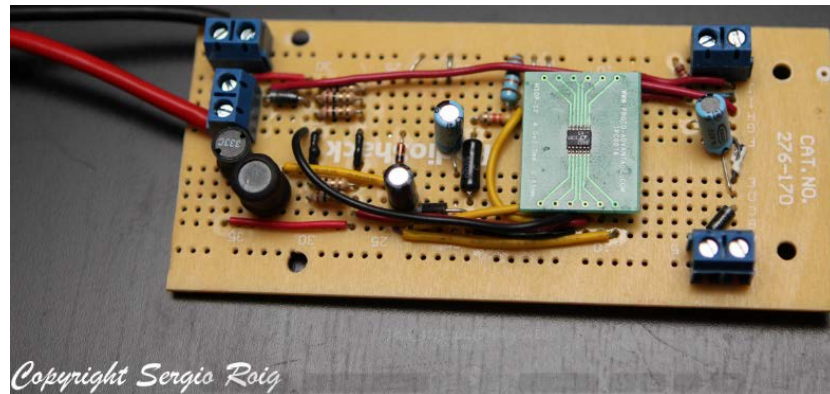


Figure 15.4.2 Charging Circuit showing LT3652 MPPT

15.5 Sensors

The hardware design wasn't significantly altered, yet we went with different components that accomplish the tasks. The barometric pressure sensor was discarded because the GPS would actually read the altitude of the plane. The single-axis and the dual-axis gyroscopes and the triple-axis accelerometer were also changed for simplicity reasons. The EAGLE libraries that we downloaded which helped in designing our PCB had different gyroscopes as well as the triple-axis accelerometer. Therefore, we swapped our components, so they fit the foot prints on our PCB. The table below shows the specifications of the accelerometer and gyroscopes.

	Triple-axis accelerometer	Single-axis gyroscope	Triple-axis gyroscope
Brand	ANALOG DEVICES	STMicroelectronics	STMicroelectronics
Part Number	ADXL335	LY530ALH	LPR53AL
Voltage Supply	1.8V – 3.6V	2.7V - 3.5V	2.7V – 3.5V
Measurement Range	+/-3.6 g	Amplified +/- 300 deg/s Not-Amplified +/- 1200 deg/s	Amplified +/- 300 deg/s Not-Amplified +/- 1200 deg/s
Sensitivity	270-330 mV/g	Amplified 3.33 mV/deg/s Not-Amplified 0.83 mv/deg/s	Amplified 3.33 mV/deg/s Not-Amplified 0.83 mv/deg/s
Size	4mmx4mmx1.45mm LFCSP	5mmx5mmx1.5mm LGA-16L	5mmx5mmx1.5mm LGA-16L

Table 15.5.1 Modified Sensors

We also changed the GPS because the one that we initially chose has an update rate of 1 Hz which was acceptable. However, our code is set for 10 Hz update rate. Therefore, we changed the Copernicus by Trimble with the MediaTek 3329 which would fix our problem. The MT3329 actually compatible with our code; its update rate is 10 Hz. Its advantage over the Copernicus is the sensitivity; the MT3329 has -165 dBm tracking sensitivity while the Copernicus has only -152 dBm tracking sensitivity. The table below shows some of the MT3329 features.

Chipset	MTK MT3329 Single chip
Channels	66
Frequency	L1, 1575,42MHz
Update Rate	10 Hz
Timing Accuracy	100ns RMS
Sensitivity	-165 dBm Tracking, -160dBm Reacquisition, -148 Acquisition
Accuracy	3m without aid
Velocity	0.1m/s without aid
Power Consumption	48mA @ acquisition, 37mA @ tracking
Dimensions	15mm x 15mm x 4mm
Weight	6 g

Table 15.5.2 Modified GPS

The hardware design after all the changes yields to the following block diagram, as shown on the figure, the barometric pressure sensor disappeared and the components are updated. The system yet functions as it was expected; hint, the GPS taking care of the altitude not the barometric pressure sensor.

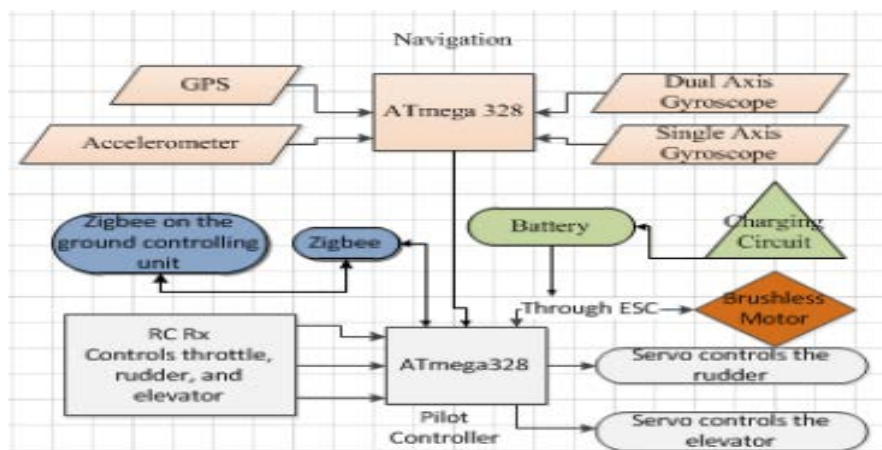


Figure 15.5.1 Modified Hardware Design

15.6 Software

Some of the changes that were made to the software portion of the project were the location of the code and the structure of the code. Plus, to keep the integrity of the code we chose not to use the waypoint function of Ardupilot. Earlier in the paper it was discussed that the code that would be used to help reduce power consumption of the UAV was to be placed in the main event loop of Ardupilot. Instead we have decided to modify one of the flight modes. The flight mode that we have that was chosen was “fly by wire b.” Below is the power saving code that was used to achieve the goal of gliding from a max altitude to a minimum altitude.

```
servo_out[CH_THROTTLE] = temp_thro;

    if(current_loc.alt < 3000){

        temp_thro = THROTTLE_MAX;

        servo_out[CH_THROTTLE] = THROTTLE_MAX;

        nav_roll = 0 ;

        nav_pitch = 1500;

    }

    if(current_loc.alt > 6000){

        temp_thro = THROTTLE_MIN;

        servo_out[CH_THROTTLE] = THROTTLE_MIN;

        nav_roll = HEAD_MAX / 3;

        nav_pitch = 500;

    }
```

The purpose of the first line of code is to keep the throttle off during the glide from the max altitude to the min altitude. Also in reverse, it keeps the throttle at a predetermined max value to return the UAV to the max altitude. The first if statement is when the aircraft will return to the max altitude. Currently it checks to see if the altitude is below 30 meters or around 100 feet and if true the UAV will straighten out while increasing its pitch. These values will be held until the plane reaches 60 meters or about 200 feet. When the UAV reaches this altitude the throttle will be cut and the pitch will be lowered till the nose of the plane is slightly elevated. The UAV will then bank to the right and gently glide down, in a circular path, to the minimum altitude.

Chapter 16 User Manual

The airframe of our project was bought from “www.nitroplanes.com” under the RC drones/UAV section. This UAV designed after the RQ-11. It comes as a frame which gave us the opportunity to install our system. The frame needed an enforced motor mount to hold a powerful motor in place. The plane originally designed to install an in-runner motor. However, we enforced the motor mount were able install an out-runner motor instead. Both in-runner and out-runner motors would fit the modification that we made to the frame. The design of the airframe requires a pusher propeller instead of a puller propeller due to having the motor mounted on the rear end of the fuselage.



Figure 16.1 AirFrame

The enforced motor mount was made out 4 circular wooden plates. They were glued together and drilled to fit the original motor mount. We used epoxy glue to stick the wooden plates together.



Figure 16.2 Fixing the motor mount

As an RC UAV we also purchased the radio controller which comes with its receiver from the same web page. The radio controller that we bought operates at 2.4GHz frequency; it is a 5-channel remote controller that allows the user to switch to several modes. We only used one channel to witch from the manual mode to the power saving mode which switches to the code that we utilized in our project. To connect the RC receiver to the autopilot board, first connect the gear channel to the control channel on the board; then connect the rudder to channel 1 on the board and connect the elevator to channel 2 on the board finally then connect throttle to channel three on the board. The figure below does have the connection shown clearly, however, the connection we

described earlier in the paragraph.

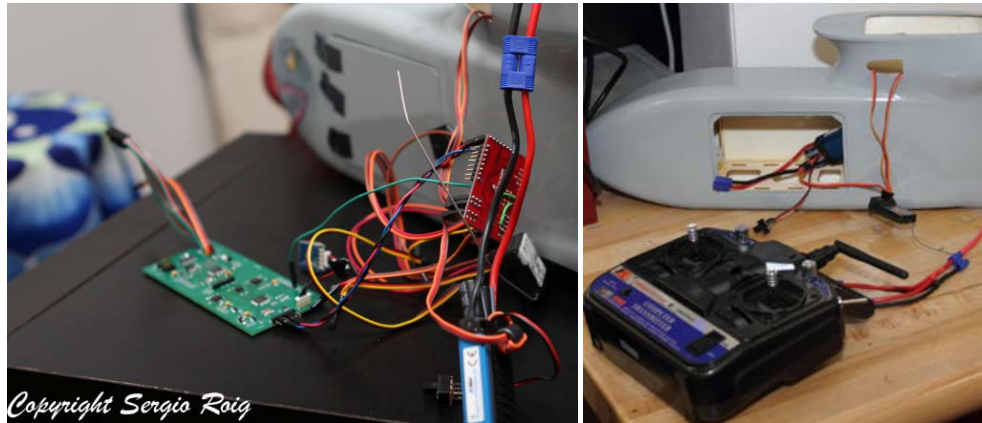


Figure 16.3 Board connections with the RC Receiver

The motor we bought is from a local hoppy shop called “Bob’s Hoppy Shop” located in Orlando, FL. It is an out-runner motor that was mounted on the outer-shell of the frame. It is an E-Flite Power-10 motor. It has 1100 Kv rating which is considered powerful enough for the Projet RQ-11. Its burst current is 42A. A 10 x 5, as the motor actually recommended, pusher propeller was used for this plane. The electronic speed controller that we bought is the E-Flite 60A Pro SB Brushless which is compatible with the E-Flite Power-10 motor. It was also purchased from a local hoppy shop called “Colonial Photo and Hoppy” which is also located in Orlando, FL. We power our circuit board via the ESC which is connected to the battery. The lithium polymer battery is charged via the charging circuit which has the maximum-power point tracker. The charging circuit is being fed by the solar panels that are installed on the wing. The autopilot PCB works according to the uploaded code which is in our case the power saving code. The PCB actually communicates with the GPS. It also communicates with Xbee at 900 MHz. Xbee communicates with the ground station by sending the throttle, altitude, the orientation of the plane, battery level and its position on Google Earth. The ground control station at it appears on the figure below shows some elements that we do not have in our design such as “AIR SPEED.” However, it is the ground station interface that we are using.

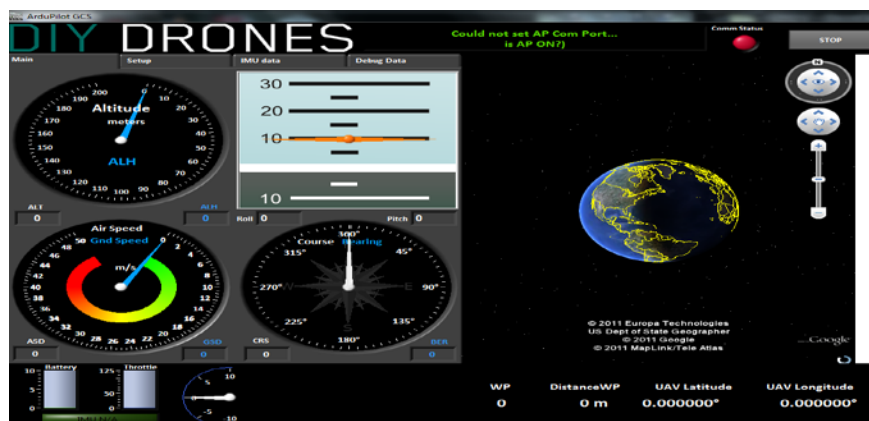


Figure 9.4.4 Ground Control Station

The solar panels are connected to the charging circuit via regular black and white wires which act as negative and positive accordingly. The charging circuit connects to the battery via a female-male Y-splitter. The other two ends of the Y-splitter one goes to charge the battery, and the other end discharges the battery to power the ESC. The ESC via an E-Flite EC3 wire powers the PCB board.

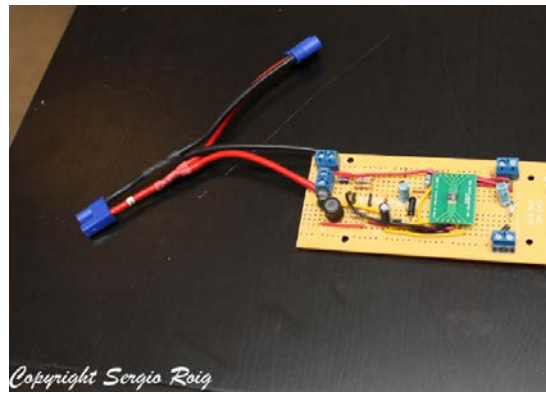


Figure 16.4 Y-Connector ESC-Battery-Charging-Circuit

Once all the connections are perfectly connected, the S.E.V. would extend its flight time by connecting the battery to the charging circuit and uploading the power saving code to the PCB. The safest way to use our system is to power up the RC controller first; the next step is to connect the battery to the ESC, turn its switch on if it has a switch. Some weight should be added at the front end of the fuselage to balance its center of gravity. The battery is placed as far as possible from the motor and the ESC which we placed to the back end of the fuselage. Next, the charging circuit and the PCB are insulated and attached on top of the battery. The RC receiver is attached to the wall of the fuselage.

The SEV is actually hand-launched and controlled via the radio controller until it reaches an altitude greater than 60 meters. Next, the user or the pilot switches one of the channels on the radio controller to operate the power-saving code. Once the code is operating the pilot keeps an eye on the ground control station until the battery is mostly used. Once the battery is 75% used the pilot has to land the SEV by switching back to the manual mode. The figure below shows the hand launch and the landing techniques.

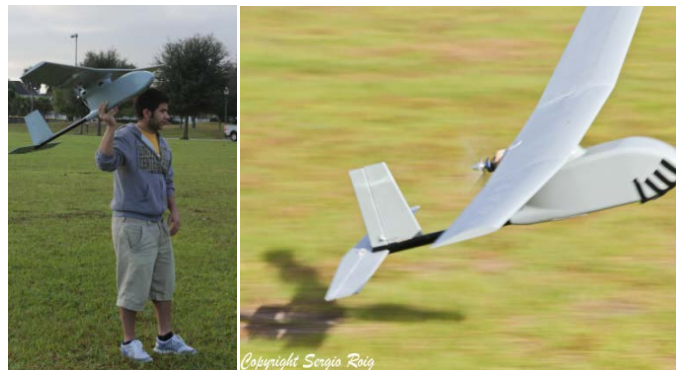


Figure 16.5 Hand-Launch / Deep-Stall Techniques

Bibliography

[1] http://www.atmel.com/dyn/products/product_card.asp?part_id=3633

[2] http://www.atmel.com/dyn/products/product_card.asp?part_id=4720

[3] http://www.atmel.com/dyn/products/product_card.asp?part_id=3618

Buchmann, Isidor. Beginners' Guide: Welcome to Model Aircraft. 14 April 2011. 28 June 2011 <<http://adamone.rchomepage.com/guide3.htm>>.

Celani, Jay. "Battery Charger's Unique Input Regulation Loop Simplifies." LT Journal of Analog Innovation (2011): 1-15.

Linear Technology . "LT3652 Power Tracking 2A Battery." Linear Technology. 10 7 2011 <<http://cds.linear.com/docs/Datasheet/3652fc.pdf>>.

Linear Technology Inc. LT3652 - Power Tracking 2A Battery Charger for Solar Power. 2010. 10 7 2011 <<http://www.linear.com/product/LT3652>>.

LiPo Wiring. 1998-2011. 5 7 2011 <http://scriptasylum.com/rc_speed/lipo.html>.

PowerFilm Solar. PowerFilm RC Aircraft Solar Module Series. 3 7 2011 <<http://www.powerfilmsolar.com/oem-components/ac-aircraft.php>>.

Silcon Solar. SolMaxx Flex 7.2V 100mA. 3 7 2011 <http://www.siliconsolar.com/flexible-solar-battery-chargers-7_2v-100ma-p-132.html>.

Silicon Solar. SolMaxx Flex 7.2V 200mA. 3 7 2011 <http://www.siliconsolar.com/flexible-solar-battery-chargers-7_2v-200ma-p-16196.html>.

Digi International Inc. "Xbee Pro 900/ DigiMesh 900 RF modules." 25 9 2009. Digi. 14 7 2011 <http://ftp1.digi.com/support/documentation/90000903_C.pdf>.

M2M Communications, An EnorNOC Company. M2MCommunications. n.d. 5 7 2011 <<http://www.m2mcomm.com/about/what-is-m2m/index.html>>.

Trimble Navigation Limited. "Copernicus GPS Receiver." 7 2006. sparkfun. 5 7 2011 <http://www.sparkfun.com/datasheets/GPS/Copernicus_Manual.pdf>.

—. "Lassen IQ GPS Receiver." 2 2005. sparkfun. 5 7 2011 <http://www.sparkfun.com/datasheets/GPS/Lassen%20iQ_Reference%20Manual.pdf>.

BOSCH. "BMP 085 Digital, barometric pressure sensor." 2 2008. bosch-sensortec. 11 7 2011 <http://www.bosch-sensortec.com/content/language1/downloads/BMP085_Flyer_Rev.0.2_March2008.pdf>.

Dimension Engineering. A beginner's guide to accelerometers. n.d. 10 7 2011 <<http://www.dimensionengineering.com/accelerometers.htm>>.

Freescale Semiconductor. " $\pm 1.5g$, $\pm 6g$ Three Axis Low-g Micromachined Accelerometer." 2010. Freescale. 10 7 2011 <http://cache.freescale.com/files/sensors/doc/data_sheet/MMA7361LC.pdf>.

InvenSense Inc. "IDG-500 Dual-Axis Gyro product specification." 20 5 2010. InvenSense. 10 7 2011 <<http://invensense.com/mems/gyro/documents/PS-IDG-0500B-00-08.pdf>>.

—. "ISZ-500 Single-axis Z-Gyro Product SpeCification." 20 5 2010. InvenSense. 20 7 2011 <<http://invensense.com/mems/gyro/documents/ps-isz-0500b-00-06.pdf>>.

Oracle ThinkQuest. BLUEPRINTS: Yaw, Pitch, and Roll. n.d. 10 7 2011 <<http://library.thinkquest.org/19952/data/english/airjumbo/blueprints/engairjumbobpypr.htm>>.

Portland State Aerospace Society. "A Quick Derivation relating altitude to air." 22 12 2004. psas.pdx.edu. 12 7 2011 <http://psas.pdx.edu/RocketScience/PressureAltitude_Derived.pdf>.

Skyhorse Publishing Inc. Glider Flying Handbook Federal Aviation Administration. New York, 2007.

Starlino. A Guid To using IMU (Accelerometer and Gyroscope Devices) in Embedded Applications. 29 12 2009. 10 7 2011 <http://www.starlino.com/imu_guide.html>.

Abbreviations

ESC	Electric Speed Controller
K_v	Motor velocity constant, measured in RPM per Volt
MPPT	Maximum Power Point Tracking
UAV	Unmanned Aerial Vehicle
mAh	Milliamp-hours
LiPo	Lithium-Polymer
NiCad	Nickel-Cadium
NiMH	Nickel-Metal Hydride
$V_{BAT(FLT)}$	Battery Float Voltage
V_{pc}	Volts per cell

Table of Figures

Figure.1 Reprinted with permission of AeroVironment Inc 1

Good morning Hamed,

You may copy any of the images on our website, but that we cannot provide high resolution versions for this purpose.

Thank you for your interest in AeroVironment, Inc.

All the best,
Cynthia

Resent-From: pr@avinc.com
From: Hamed Alostath <halostath@knights.ucf.edu>
Date: November 13, 2011 12:50:38 PM PST
To: <pr@avinc.com>, <evscs@avinc.com>
Subject: Senior Design Reprinting Request

Cynthia Nichols
AeroVironment, Inc.
Executive Assistant
Corporate

nichols@avinc.com

Tel: (626) 357-9983 ext. 510
Fax: (626) 359-9628

NASDAQ: AVAV
www.avinc.com

CONFIDENTIALITY STATEMENT: This electronic message transmission contains information from AeroVironment, Inc. and is confidential and/or proprietary. The information is intended only for the use of the individual or entity named above. If you are not the intended recipient, be aware that any disclosure, copying, distribution or use of the contents of this message is prohibited. If you have received this electronic transmission in error, please send a reply notification and/or notify us by telephone (626-357-9983) immediately and delete this message from your computer.

Hello,

I'm an electrical engineering student at the University of Central Florida. I am working on my senior design project - solar powered autonomous sailplane - which needs documentation. I am actually interested in including your Raven B system in the paper to enhance my senior design documentation. therefore, I am requesting your permission to use some of the images you have online.

Sincerely,
Hamed Alostath
halostath@knights.ucf.edu

Figure 2.3.1 4
Figure 3.2.1 Apprentice 15E "Reprinted with the permission of E-flite®" 7

Dear Hamed,

Thank you for your interest in our products. Permission has been granted to use the image of the Apprentice 15E on our website. Good luck with your project!

Photo Studio
Horizon Hobby, Inc.

-----Original Message-----

From: halostath@knights.ucf.edu [mailto:halostath@knights.ucf.edu]
Sent: Sunday, November 13, 2011 2:56 PM
To: Horizon Hobby Sales
Subject: Contact from E-flite - Advancing Electric Flight.com

Comments: Hello

I am an electrical engineering student at the University of Central Florida. I am actually working on my senior design project - Solar Power Sailplane. I am requesting your permission to use the Apprentice 15E image that you have online to enhance my documentation of the project. I am also using it for testing my system.

Sincerely,
Hamed Alostath
halostath@knights.ucf.edu

Figure 3.2.2 The Radian "Reprinted with the permission of Parkzone®".....8
Dear Hamed,

Thank you for your interest in our products. Permission has been granted to use the image of the Radian on our website. Good luck with your project!

Photo Studio
Horizon Hobby, Inc.

-----Original Message-----

From: halostath@knights.ucf.edu [mailto:halostath@knights.ucf.edu]
Sent: Sunday, November 13, 2011 3:01 PM
To: Horizon Hobby Sales
Subject: Contact from New HorizonHobby.com

Comments: Hello,

I am an electrical engineering student at the University of Central Florida. I am working on my senior design project - Solar power Sailplane. I am requesting your permission to use the Radian image that you have online in my paper to enhance my documentations of the project.

Sincerely,
Hamed Alostath
halostath@knights.ucf.edu

Figure 3.3.1 RQ-11 "Reprinted with permission of AeroVironment Inc"(shown above)...9
Figure 3.8.1 ESC Functions.....15
Figure 4.1.1 ATmega328 "reprinted with permission of Atmel".....16

From: webmaster-bounces@atmel.com [mailto:webmaster-bounces@atmel.com] **On Behalf Of** Hamed Alostath
Sent: Friday, July 22, 2011 2:11 PM
To: webmaster@atmel.com
Subject: [Webmaster] Reprinting Request for Senior Design Project

Hello,

I am an electrical engineering student at University of Central Florida; I am working on my senior design project which is a solar powered autonomous sailplane. As I am searching for the best micro-controller that fits my project, I found out that ATmega 323 is the one I need. Therefore, I am requesting to use in my research paper the following: the pin configurations, some of the graphs, and some the tables which are offered in its online user manual.

Sincerely,

Hamed Alostath
halostath@knights.ucf.edu
Hamed,

You may use the materials you've described. The only criteria is that Atmel is cited as the source.

I am copying Shveta Miglani who helps manage our university program and might be interested in learning more about your senior project.

Good luck on your project.

Regards,

Michael De Caro
Senior Web Operations Manager / Atmel Corporation
Tel: (+1) (408) 436-4352 / Fax: (+1) (408) 487-2600
michael.decaro@atmel.com / www.atmel.com

The information contained in this email message may be privileged, confidential and/or protected from unauthorized disclosure. If you are not the intended recipient, any dissemination, distribution or copying is strictly prohibited. Please immediately notify the sender by reply if you received this email in error. Thank you for your cooperation.

Figure 4.4.1 Layout of the autopilot board.....	20
Figure 4.4.2 the charging circuit PCB.....	21
Figure 4.6.1 inside the fuselage of the UAV.....	22
Figure 4.6.2 the battery back pack mounting of the PCBs.....	23, 101
Figure 5.2.1: ATmega1280 "Reprinted with the permission of Atmel".....	26
Figure 5.2.2: ATmega328 "Reprinted with the permission of Atmel".....	27
Figure 5.3.1.....	28
Figure 5.3.2.....	29
Figure 6.1 Roll, Pitch, and Yaw. "Reprinted with the permission of "AeroVironment" (shown above).....	31
Figure 6.2 the ISZ-500 and IDG-500 Integration.....	33
Figure 6.3 Accelerometer Functionality.....	34
Figure 6.4 MML7361L and ISZ-500 IDG-500 integration.....	35
Figure 6.5 the integration of all the sensors.....	37
Figure 7.1 Machine to Machine Communications.....	38
Figure 7.2 GPS integration.....	41
Figure 7.3 the connections with the microcontroller.....	43
Figure 8.1 charging circuit overview.....	44

Figure 8.2.2.1.....47
 Figure 8.2.2.2.....47
 Figure 8.2.2.3.....48, 90
 Figure 8.2.3.1 Solar Array Configuration.....48, 91
 Figure 8.3.3.1.....51
 Figure 8.4.1.1 “Reprinted with permission of Linear Technology”52

Sergio Roig

From: bob.dobkin [rdobkin@linear.com]
Sent: Thursday, July 07, 2011 4:34 PM
To: John Hamburger; serge74@knights.ucf.edu
Subject: Re: Senior Design Image Reprinting

You are authorized to reprint information from Linear datasheets. If you extract figures, please include "courtesy Linear Technology Corporation"

Robert Dobkin
 VP/CTO

*****LINEAR TECHNOLOGY CORPORATION*****
 *****Internet Email Confidentiality Footer*****
 This e-mail transmission, and any documents, files or previous e-mail messages attached to it may contain confidential information that is legally privileged. If you are not the intended recipient, or a person responsible for delivering it to the intended recipient, you are hereby notified that any disclosure, copying, distribution or use of any of the information contained in or attached to this transmission is STRICTLY PROHIBITED. If you have received this transmission in error, please immediately notify me by reply e-mail, or by telephone at (408) 432-1900, extension 3751 and destroy the original transmission and its attachments without reading or saving in any manner. Thank you.
 =====

On Jul 7, 2011, at 9:52 AM, John Hamburger wrote:

Sergio Roig [<mailto:serge74@knights.ucf.edu>]

Figure 8.4.2.1 “Reprinted with permission of Linear Technology” (shown above).....53
 Figure 8.4.3.1 “Reprinted with permission of Linear Technology” (shown above).....54
 Figure 8.4.4.1 “Reprinted with permission of Linear Technology” (shown above).....55
 Figure 8.4.4.2 Programming Maximum Charge Current Using R_{SENSE} “Reprinted with permission of Linear Technology” (shown above)55
 Figure 8.4.4.3 Feedback Resistors from BAT to V_{FB} Programming Float Voltage “Reprinted with permission of Linear Technology” (shown above).....56
 Figure 8.4.4.4 Resistor Divider Sets Minimum V_{IN} “Reprinted with permission of Linear Technology” (shown above).....56
 Figure 8.4.4.5 Charging Circuit Design.....58, 94
 Figure 9.2.1 Main Loop60, 96
 Figure 9.2.2 Fast Loop Function.....60
 Figure 9.2.3 Medium Loop Function.....61

Figure 9.2.4: The main loop event for allowing the UAV to glide.....63, 97
 Figure 9.2.5: Visual representation of the main loop event64
 Figure 9.4.1: Mission Planner with home location at Bright House Stadium.....65
 Figure 9.4.2: Mission Planner with waypoints selected66, 98
 Figure 9.4.4: Ground Control Station67,122
 Figure 10.1 Hardware Design69
 Figure 10.2 Pin Configurations. “Reprinted with the permission of Atmel” (shown above)
71
 Figure 10.3 Navigation Unit73
 Figure 10.4 Pin Configurations. “Reprinted with the permission of Freescale”74

From: Hamed Alostath [mailto:halostath@knights.ucf.edu]
Sent: Tuesday, July 26, 2011 5:52 AM
To: McLellan John-R2AAKL
Subject: Reprinting Request for Senior Design Project

Dear John McLellan,

I am an electrical engineering student at University of Central Florida; I am working on my senior design project which is a solar powered autonomous sailplane. As I am searching for the best accelerometer that fits my project, I found out that MMA7361L is the one I need. Therefore, I am requesting to use in my research paper some of the figures, which are used in its technical data sheet, such as: the pin configurations and their descriptions, in addition, to its connection with the microcontroller, and some of the tables which are offered online by Freescale.

Sincerely,

Hamed Alostath
 halostath@knights.ucf.edu

7/26/11
Reply

□ McLellan John-R2AAKL [Add to contacts](#)
 To: Hamed Alostath, Mastronardi Andy-RAT119

From: **McLellan John-R2AAKL** (r2aakl@freescale.com)
 Sent: Tue 7/26/11 12:33 PM
 To: Hamed Alostath (halostath@knights.ucf.edu)
 Cc: Mastronardi Andy-RAT119 (RAT119@freescale.com)

Hamed,

I am assuming this is publicly available information. Typical fair use is as long as you are NOT using our copyright material for commercial purposes and you site Freescale as the source of any figures or tables used.

Regards,

~John McLellan~
 University Programs
 Freescale Semiconductor Inc.
 512.895.7285
johnmc@freescale.com

Figure 10.5 Recommended interfacing accelerometer to microcontroller. “Reprinted with the permission of Freescale” (shown above).....75
 Figure 10.6 Pin Configurations and Axes Representation. “Reprinted with the permission of InvenSense” (pending)77

Reprint Request for Senior Design Project

To see messages related to this one, [group messages by conversation](#).

Full view

Hamed Alostath

To sales@invensense.com

7/30/11

Reply

From: **Hamed Alostath** (halostath@knights.ucf.edu)
Sent: Sat 7/30/11 3:41 AM
To: sales@invensense.com

Hello

I am an electrical engineering student at University of Central Florida; I am working on my senior design project which is a solar powered autonomous sailplane. As I am searching for the best gyroscopes that fit my project, I found out that the IDG-500 and ISZ-500 are the ones I need. Therefore, I am requesting to enrich my research paper by using some of the figures, which are used in their specifications sheets, such as: the pin configurations and their descriptions, in addition to some of the tables which are offered online by InvenSense.

Sincerely,

Hamed Alostath
halostath@knights.ucf.edu

Figure 10.7 IDG-500 Reference Circuit. “Reprinted with the permission of InvenSense”(pending) (shown above)78
Figure 10.8 BMP085 integration. “Reprinted with the permission of BOSCH”80

```
subject: Product-related contact form submission
email: halostath@knights.ucf.edu
reqdata: datasheet
template: product-contact.txt
city:
redirect: ../../content/language1/html/3792.htm?sentform=1
jobtype: others
title:
company: A student at University of Central Florida
faxnumber:
firstname: Hamed
lastname: Alostath
state: Florida
phonenummer:
street:
zip: 32828
country: USA
product: pressure_sensors
requestedinformation: For the purpose of graphs, and tables that are offered by BOSCH. more
specifically. I chose to use the BMP085 digital pressure sensor on my project. therefore, to
enrich my design, I am requesting to make use of its data sheet.
purpose: I am working on my senior design project which is a solar powered sailplane. the
pressure sensor is going to be used to enhance the sensitivity of the plane
Dear Mr. Alostath,

Thank you for the request. It is interesting to hear that you will use our sensor in your
project.

The data sheet is available on the internet and can be used. (http://www.bosch-
sensortec.com/content/language1/html/3477.htm)
Please include the reference "Picture: Bosch" if you use images out of it. For PR pictures,
please take a look at http://www.bosch-presse.de/presseforum/search.htm?
query=sensortec&type=image. Here you can find the pictures and press releases of our sensors.

Mit freundlichen Grüßen / Best regards

Martina Förster

Bosch Sensortec GmbH
Marketing Communication (BST/MKT)
Gerhard-Kindler-Straße 8
72770 Reutlingen
GERMANY
www.bosch-sensortec.com
```

Figure 10.9 Copernicus™ Pin Configurations and minimum Antenna Connections.
 “Reprinted with the permission of Trimble”82

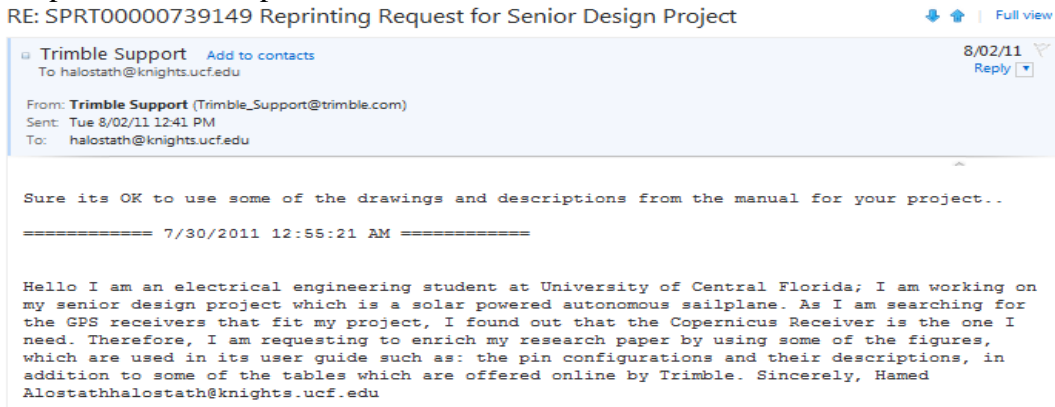


Figure 10.10 the Pilot Controller Unit83, 90
 Figure 10.11 the Pilot Controller Integration84
 Figure 10.12 Data Communication84
 Figure 10.13 The Internal Flow Diagram. “Reprinted with the permission of Digi”
86

RE: Digi Service Case: 1318945 Reprinting Request for Senior Design Project

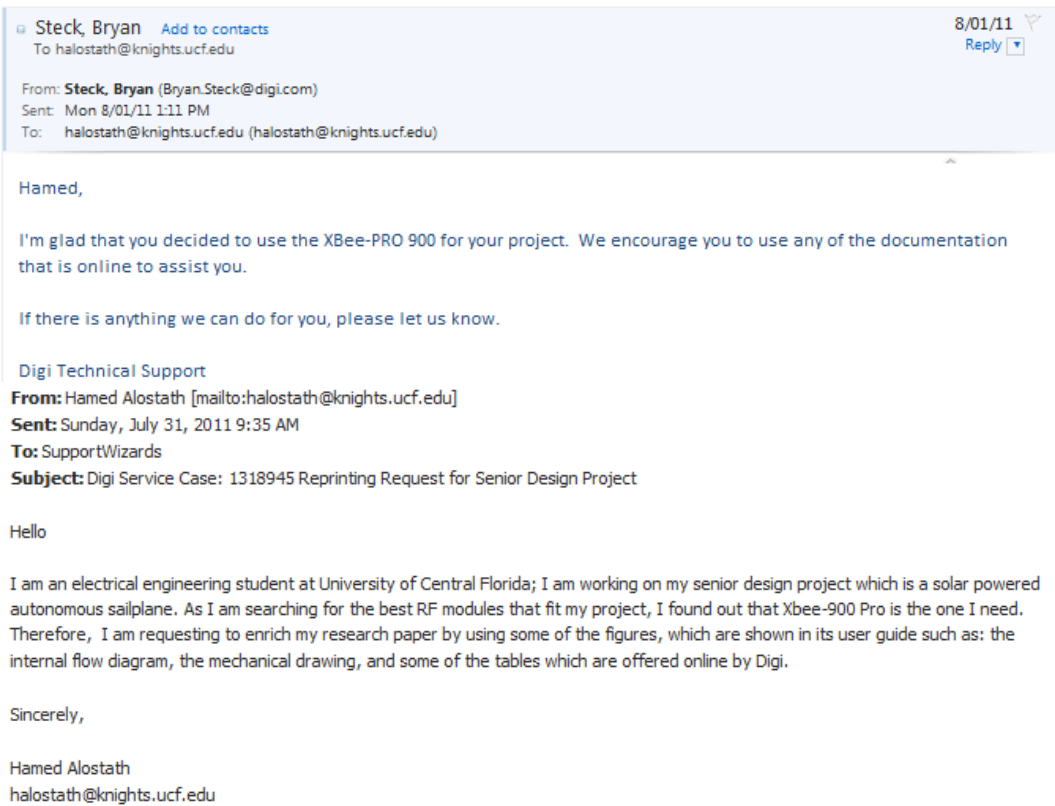


Figure 11.1 "Reprinted with the permission of Nitroplanes"87

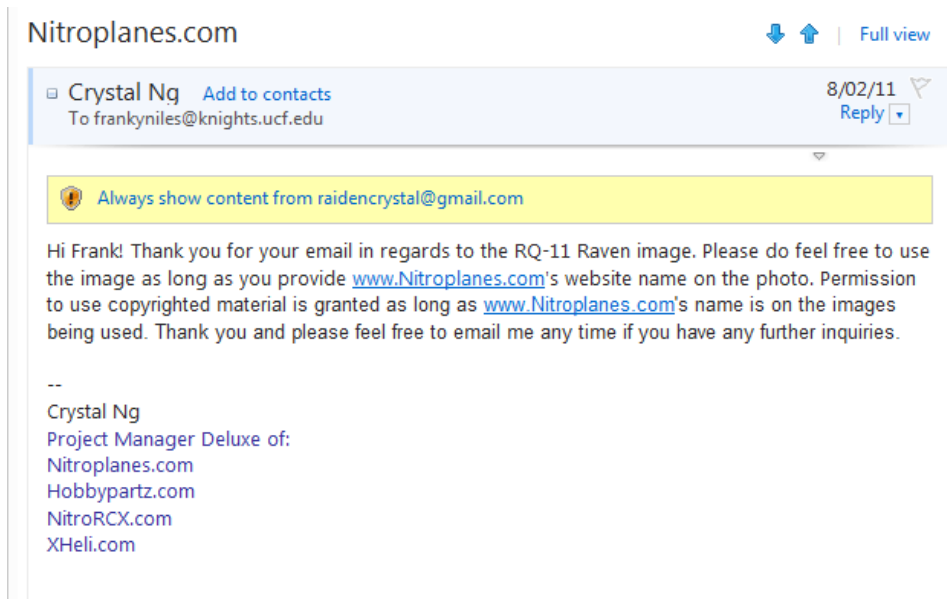


Figure 11.2 the communication System Design Summary95

Figure 11.3 Block Diagram99

Figure 12.1.1 Inside the fuselage of the UAV100

Figure 12.1.2 The battery back pack mounting of the PCBs101

Figure 12.1.3 Assorted parts of the RQ-11102

Figure 12.1.4 - Main wing of the RQ-11103

Figure 12.1.5- Partially built RQ-11 airframe103

Figure 12.1.6 - Top view of the prop tower104

Figure 12.1.7 - Bottom view of the fuselage104

Figure 12.1.8 – Tailfin of the RQ-11105

Figure 13.2.1- Dev boards purchased from Sparkfun.com107

Figure 13.5.1 flight path109

Figure 14.3.1 Summer 2011 Gantt Chart111

Figure 14.3.2 Fall 2011 Gantt Chart112

Figure 15.1.1 Motor mount before and after enforcement.....114

Figure 15.2.1 Power-10 Motor.....115

Figure 15.3.1 The E-Flite 60A.....116

Figure 15.4.1 The Autopilot Board.....117

Figure 15.4.2 The Charging-Circuit Board.....118

Figure 15.5.1 Modified Hardware Design.....119

Figure 16.1 Airframe.....121

Figure 16.2 Fixing the Motor Mount.....121

Figure 16.3 Board Connections with the RC Receiver.....122

Figure 16.4 Y-connectro ESC-Battery-Charginc-Circuit.....123

Figure 16.5 Hand-Launch/Deep-Stall Techniques.....123

Table of Tables

Table 3.5.1 Motor spec comparison.....12
 Table 3.6.1 Servo spec comparison.....13
 Table 3.7.1 ESC spec comparison.....14
 Table 4.2.1 Comparison of features between different PCB vendors17, 18
 Table 6.1 General Requirements for the Gyro31
 Table 6.2 ISZ-500 and IDG-500 Specifications32
 Table 6.3 MMA7361L basic requirements34, 35
 Table 6.4 symbol descriptions according to Portland State Aerospace Society36
 Table 6.5 Basic Requirements37
 Table 7.1 GPS General Requirements40
 Table 7.2 Copernicus™ GPS vs. Lassen iQ GPS41
 Table 7.3 Condition of Logic Signals42
 Table 7.4 Basic Requirements of the Zigbee43
 Table 8.2.2.1 Solar panel comparison.....46, 47
 Table 8.4.4.1 Resistors/Inductor calculations.....57, 93
 Table 8.4.4.2 Status pins state“Reprinted with permission of Linear Technology” ..57, 94

Sergio Roig

From: bob dobkin [rdobkin@linear.com]
Sent: Thursday, July 07, 2011 4:34 PM
To: John Hamburger; serge74@knights.ucf.edu
Subject: Re: Senior Design Image Reprinting

You are authorized to reprint information from Linear datasheets. If you extract figures, please include "courtesy Linear Technology Corporation"

Robert Dobkin
 VP/CTO

*****LINEAR TECHNOLOGY CORPORATION*****
 *****Internet Email Confidentiality Footer*****
 This e-mail transmission, and any documents, files or previous e-mail messages attached to it may contain confidential information that is legally privileged. If you are not the intended recipient, or a person responsible for delivering it to the intended recipient, you are hereby notified that any disclosure, copying, distribution or use of any of the information contained in or attached to this transmission is STRICTLY PROHIBITED. If you have received this transmission in error, please immediately notify me by reply e-mail, or by telephone at (408) 432-1900, extension 3751 and destroy the original transmission and its attachments without reading or saving in any manner. Thank you.
 =====

On Jul 7, 2011, at 9:52 AM, John Hamburger wrote:

Sergio Roig [<mailto:serge74@knights.ucf.edu>]

Table 10.1 Autopilot System68

Table 10.2 Pin Descriptions according to Atmel’s user manual	70
Table 10.3 Components Analysis	72
Table 10.4 MMA7361L Pin Descriptions	73, 75
Table 10.5 MMA7361L Connections	75
Table 10.6 Pin Descriptions of the IDG-500	76
Table 10.7 IDG-500 Connection	77
Table 10.8 ISZ-500, IDG-500, and MMA7361L Connection	79
Table 10.9 BMP085 Pin connections	80
Table 10.10 the GPS Pin Descriptions	81
Table 10.11 Copernicus™ GPS Connections	82
Table 10.12 Zigbee Pro pin Descriptions	85
Table 10.13 ZigbeePro Connections	86
Table 11.1 Motor Specs.....	88
Table 11.2 Servo Specs.....	88
Table 11.3 ESC Specs.....	89
Table 14.1.1 Budget and Financing	110
Table 15.2.1 Power-10 Specifications.....	115
Table15.3.1 E-Flite 60-A Pro SB Brushless Specifications.....	116
Table15.5.1 Modified Sensors.....	118
Table 15.5.2 Modified GPS.....	119